Bilingual Handwritten Indian language translation

TEAM MEMBERS

1) 21BCE7756 -T.MOHITH
2)21BCE9862 - D.MOUNIKA

AGENDA

1. Introduction
2. Problem Statement
3. Dataset
4. Preprocessing
5. Model Architecture
6. Evaluation Metrics
7. Results and Observations
8. Conclusion

PROBLEM STATEMENT

Sanskrit, one of the world's oldest and most complex languages, holds vast knowledge within its ancient texts. Its intricate grammar, rich syntax, and diverse vocabulary make translation to English particularly challenging, especially with handwritten texts due to varying writing styles.

This project aims to develop a robust system for translating handwritten Sanskrit into English using advanced deep learning techniques, specifically LSTM networks. By addressing these challenges, the project seeks to preserve cultural heritage, enable academic research, and advance language processing technologies.

DATA PREPARATION

a)Data Collection: Gather Sanskrit images and corresponding English translations from digital archives, research repositories, and open-source datasets.Ensure diversity in handwriting styles and document types for better generalization.

b)Data Preprocessing: Preprocessing involves cleaning and preparing the data for model training. First, duplicates are removed to avoid redundancy. Next, missing values are handled by removing incomplete data pairs. Sentences are then tokenized into words or subwords, followed by padding to ensure all sequences are of equal length.

c) Data Splitting: Split the dataset into training (80%) and test (20%) sets to balance the evaluation process.

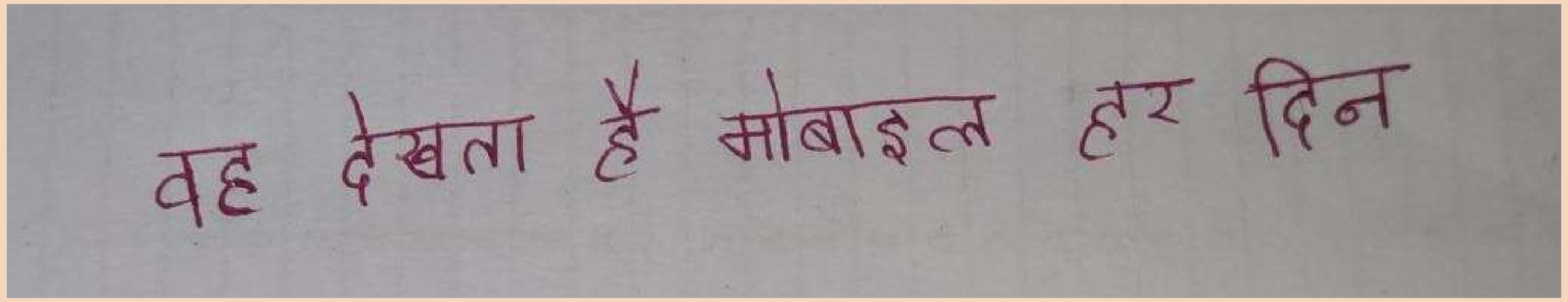
Data Set

id	Sanskrit	English
c:1v1	धृतराष्ट्र उवाच |धर्मक्षेत्रे कुरुक्षेत्रे समवेता युयुत्सवः |मामकाः पाण्डवाश्चैव किमकुर्वत सञ्जय ||	Dhritarashtra said: O Sanjay, after gathering on the holy field of Kurukshetra, and desiring to fight, what did my sons and the sons of Pandu do?
c:1v2	सञ्जय उवाच ।दृष्ट्वा तु पाण्डवानीकं व्यूढं दुर्योधनस्तदा । आचार्यमुपसङ्गम्य राजा वचनमब्रवीत् ।। ।।	Sanjay said: On observing the Pandava army standing in military formation, King Duryodhan approached his teacher Dronacharya, and said the following words.
c:1v3	पश्यैतां पाण्डुपुत्राणामाचार्य महतीं चमूम् ।व्यूढां द्रुपदपुत्रेण तव शिष्येण धीमता ।। ।।	Duryodhan said: Respected teacher! Behold the mighty army of the sons of Pandu, so expertly arrayed for battle by your own gifted disciple, the son of Drupad.
c:1v4	अत्र शूरा महेष्वासा भीमार्जुनसमा युधि	Behold in their ranks are many powerful warriors, like Yuyudhan, Virat, and Drupad, wielding mighty bows and equal in military prowess to Bheem and Arjun. There are also accomplished heroes like Dhrishtaketu, Chekitan, the gallant King of Kashi, Purujit, Kuntibhoj, and Shaibya—all the best of men. In their ranks, they also have the courageous Yudhamanyu, the gallant Uttamauja, the son of Subhadra, and the sons of Draupadi, who are all great warrior

अनन्तविजयं राजा कुन्तीपुत्रो युधिष्ठिरः ।नकुलः सहदेवश्च सुघोषमणिपुष्पकौ ।।

काश्यश्च परमेष्वासः शिखण्डी च महारथः ।धृष्टद्युम्नो विराटश्च सात्यकिश्चापराजितः ।।

द्रुपदो द्रौपदेयाश्च सर्वशः पृथिवीपते ।सौभद्रश्च महाबाहुः शङ्खान्दध्मुः पृथक् पृथक् ।।



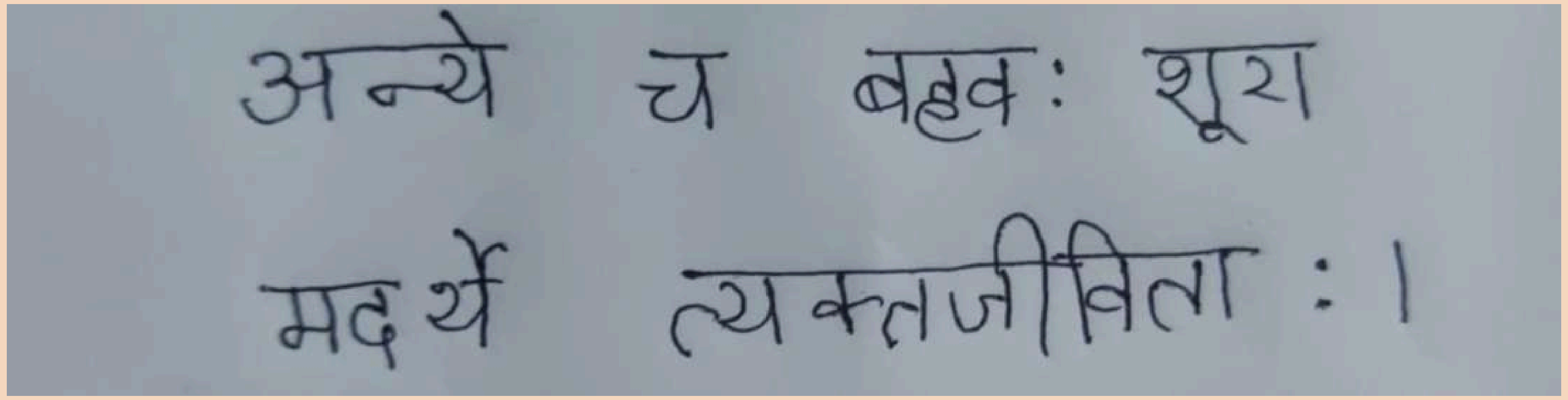


IMAGE PROCESSING

a) Handwritten Text Recognition:

Preprocessing images: 1) Convert to grayscale to reduce computational complexity.

2) Perform noise reduction using median filters.

3)Does De-skewing: Corrects any rotation or skew present in the text, ensuring the text is properly aligned horizontally and the contrast enhancement

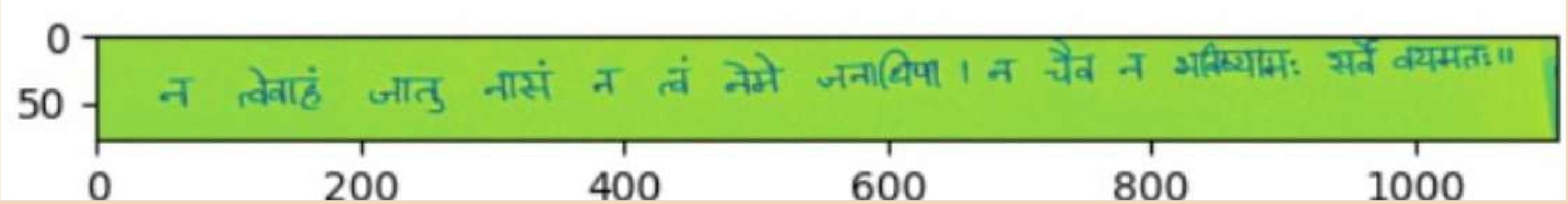
b) Output Text:

Convert recognized Sanskrit characters into digital text format to feed into the translation model.

Your paragraph text

Text Extraction

IMAGE 1:



Text from Image 1:
न त्वं नेमे जनाधयिया न चैव न भविष्यामः   सर्वे 4यमतः"१ न वेवाहं ज्◌ातु नासं

IMAGE 2:

यं हिन व्यथयन्त्येतें समदुःस्सुञ्जं धीरं सोडमृतत्वाय कल्पते || पुरुषं पुरुषषभ

MODEL DEVELOPMENT

a)Embedding Layer: Create Sanskrit-specific embeddings to capture semantic relationships between words.
Add positional encoding to retain the order of words in Sanskrit, crucial for grammatical accuracy.

b)Encoder-Decoder Architecture:

Encoder: A bidirectional LSTM processes the input Sanskrit sequence, capturing both past and future context for each word.
Attention Mechanism: Use attention to focus on specific Sanskrit words that are most relevant for generating the current word in the English translation.
Decoder: Another LSTM generates English words one at a time based on the encoder's context and attention outputs.

c)Regularization:

Apply dropout to LSTM layers to prevent overfitting.
Use gradient clipping to stabilize training and avoid exploding gradients.

d)Hyperparameter Optimization:

Experiment with learning rates, batch sizes, and dropout rates to achieve optimal model performance

DEEP LEARNING INVOLVEMENT

Translation Model:
The translation model for Sanskrit-to-English uses a Seq2Seq architecture with LSTM cells and an Attention Mechanism to handle the complexities of both languages. The Encoder-Decoder model processes the input Sanskrit sentence, where the encoder (using LSTM) creates a context vector, and the decoder generates the English translation. To improve translation accuracy, the Attention Mechanism allows the model to focus on relevant parts of the input sentence at each decoding step, enhancing context retention for long sentences.

Your paragraph text

LSTM

WHAT IS LSTM?

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to process sequential data by remembering long-term dependencies.

Unlike other types of RNNs, LSTMs directly address the vanishing gradient problem, making them well suited to handle long sequences, which is important for languages like Sanskrit.

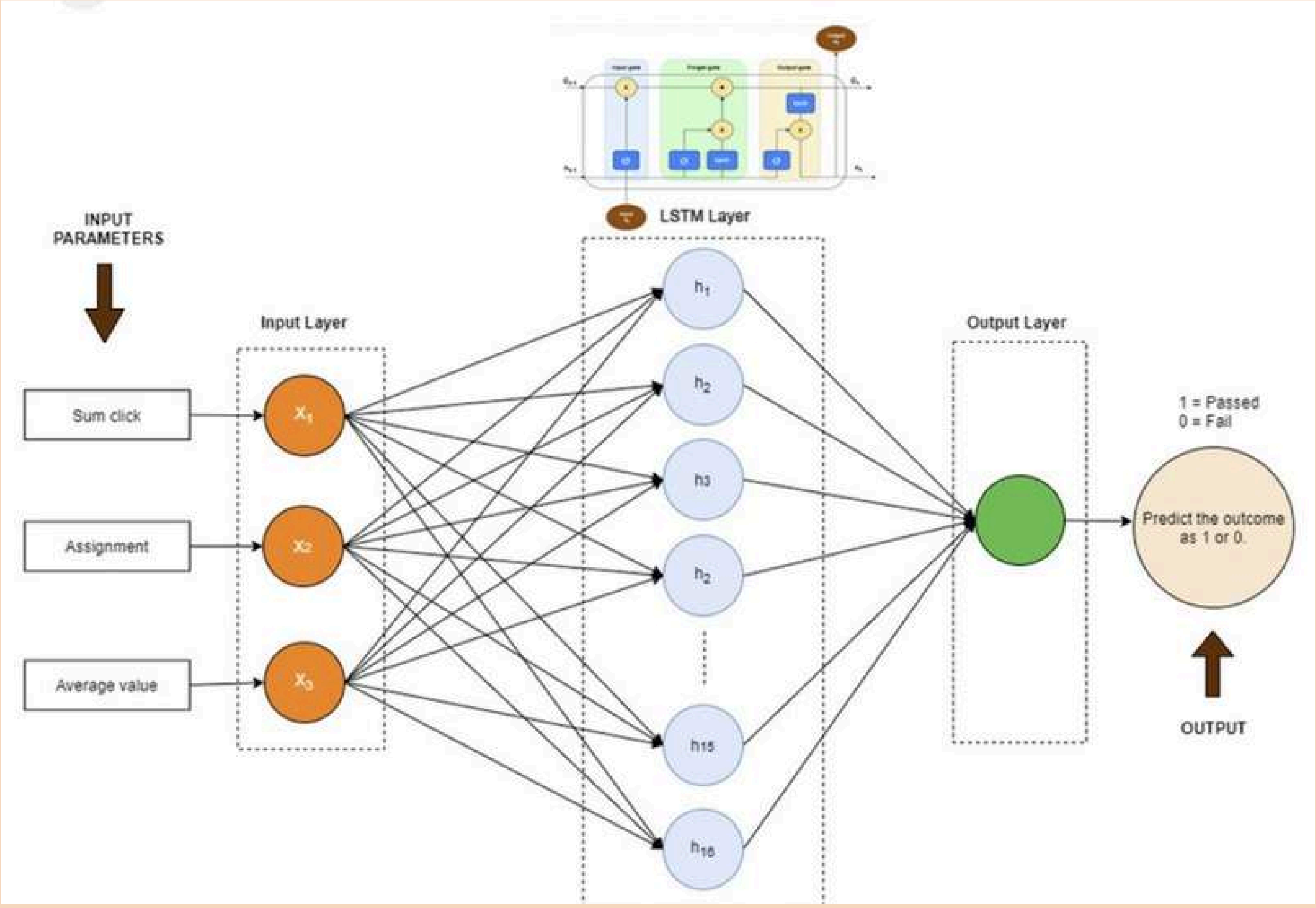
LSTM for Sanskrit Translation:

a)Sequential Data Handling: Sanskrit has flexible syntax and long sequences of words, which LSTM can effectively model.

b)Context Preservation: LSTM's memory cells store past information, capturing the contextual relationships between Sanskrit words for accurate translation.

c)Adaptability to Complex Grammar: Handles the intricate grammatical structures of Sanskrit, such as compounds (sandhi) and case-based relationships.

Workflow of LSTM



LSTM Workflow for Translation:

1)Encoder-Decoder Architecture: Encoder LSTM: Processes the Sanskrit input sequence and encodes it into a context vector. Decoder LSTM: Generates the English translation based on the context vector. **2)Attention Mechanism:** Adds focus on specific words in the Sanskrit sentence while decoding, improving translation accuracy. 3)Training with Sanskrit-Specific Features: Preprocessed parallel Sanskrit-English datasets are fed into the model. Custom embeddings capture unique Sanskrit linguistic properties.

Sequence-to-Sequence (Seq2Seq) Model

WHAT IS Seq2Seq:

A Sequence-to-Sequence (Seq2Seq) model is a type of deep learning architecture that transforms one sequence into another sequence. It is widely used for tasks like machine translation, text summarization, and speech recognition, where the input and output data are sequential but may differ in length and structure.

Components of a Seq2Seq Model:

Encoder

The encoder processes the input sequence (e.g., a Sanskrit sentence) step-by-step, encoding it into a fixed-length context vector that represents the semantic and syntactic meaning of the input. It is typically implemented using layers like LSTM which excel at handling sequential data.

Example:

Sanskrit Input: रामः ग त व ालयम्।

Decoder

The decoder generates the output sequence (e.g., English translation) step-by-step using the context vector from the encoder and previously generated words. It is commonly implemented with LSTMsfor accurate sequential predictions.

Example:

Output Sequence: Rama goes to school.

How the Seq2Seq Model Works

1)Input Processing (Encoder):
The encoder takes the input sequence (e.g., a Sanskrit sentence) and processes it step-by-step. It converts the sequence into a context vector, which summarizes the entire input's semantic and syntactic meaning. This vector acts as a bridge between the input and output sequences.

2)Translation (Decoder):
The decoder generates the output sequence (e.g., an English translation) by interpreting the context vector. It produces words one at a time, ensuring the output aligns with the meaning captured in the context vector. The decoder ensures coherence by maintaining the sequential flow of words.

3)Feedback Loop:
During training, the decoder often uses teacher forcing, where the correct word from the target sequence is fed back as input for the next step, improving learning. In real-time translation, the decoder uses its previously generated word as input for predicting the next word. This process ensures dynamic, step-by-step translation.

TRAINING

Train the model on the Sanskrit-English training dataset (80%) using LSTM and attention mechanisms, monitoring loss and accuracy.

Optimize hyperparameters like learning rate and dropout to improve performance and reduce overfitting.

TESTING

Evaluate the model on the test dataset (20%) .

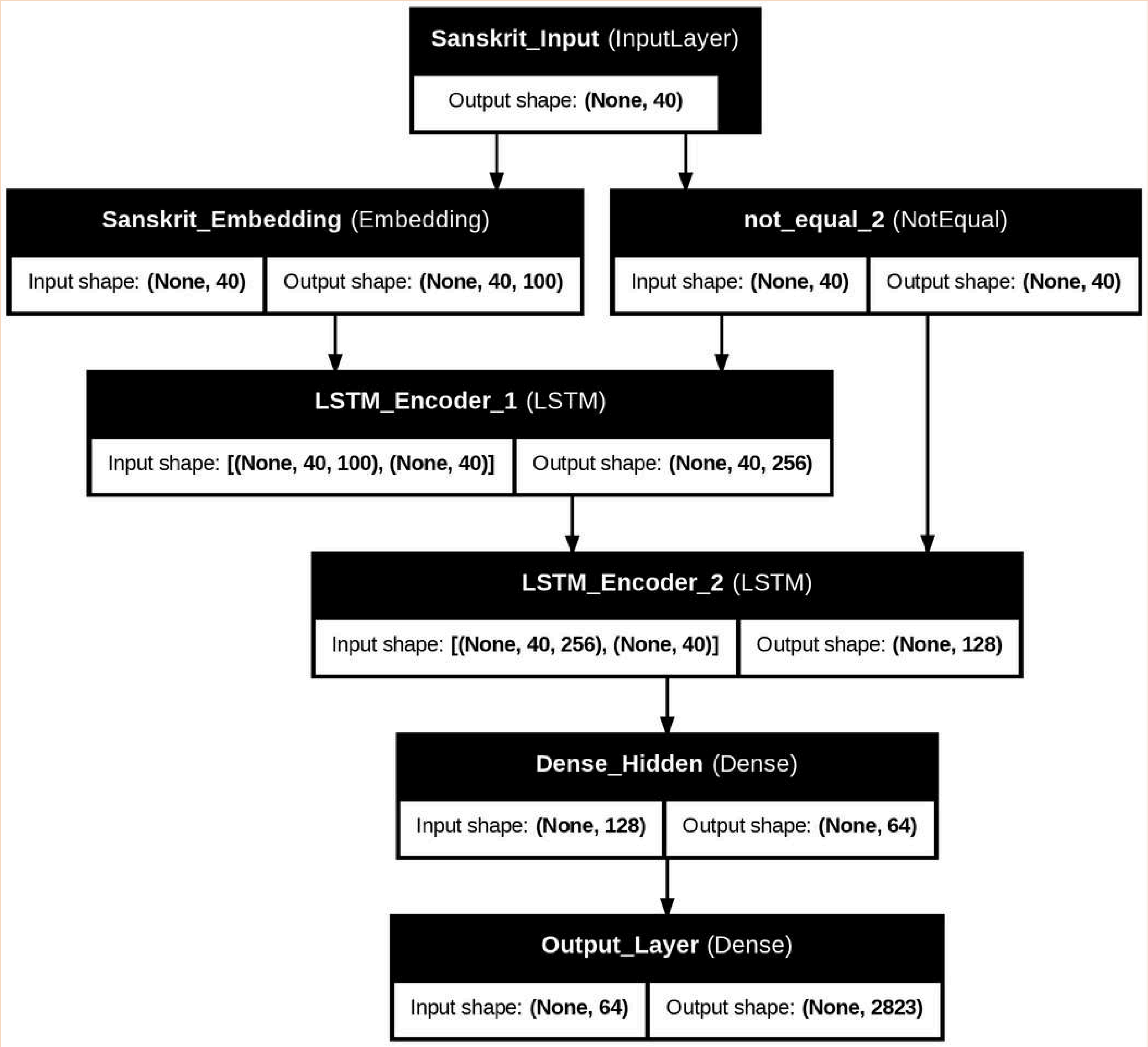
Test robustness with real-world handwritten Sanskrit texts and analyze error patterns.

DEPLOYMENT

1)Develop a user-friendly interface for text input (image) and display translated English output.
2)Deploy the system on the cloud for accessibility and scalability.
3)Conduct user testing to gather feedback for improvements.

Your paragraph text

OUR MODEL ARCHITECTURE



RESULT

```
Test Loss: 0.0979
Test Accuracy: 99.29%
```

PRECISION:

Precision measures the proportion of correctly predicted positive results (words) out of all predicted positive results.

Precision=True Positives (TP)/False Positives (FP)+True Positives (TP)

Precision in translation refers to the proportion of words in the predicted translation (English) that are correct. A higher precision means that the model’s translations are accurate in terms of the words it predicts, but it may not be complete or may miss some important words.

PRECISION: 0.9858

RECALL:

Recall measures the proportion of correctly predicted positive results (words) out of all actual positive results (words) that should have been predicted.

Recall=True Positives (TP)/False Negatives (FN)+True Positives (TP)

Recall in translation refers to how many of the correct words in the target language (English) the model was able to predict. High recall means the model generates many correct words, but it may also generate irrelevant or incorrect words (false positives).

Recall:0.9929

F1 SCORE:

The F1 Score is the harmonic mean of Precision and Recall. It is a measure of a test's accuracy and is particularly useful when you want to balance Precision and Recall.

The formula is:

F1=2 · Precision.Recall/Precision +Recall

F1 :0.9893

PRECISION ,RECALL ,F1,ROUGE SCORE

```
Precision: 0.9858
Recall: 0.9929
F1 Score: 0.9893

Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       139
         254       0.00      0.00      0.00         1

    accuracy                           0.99       140
   macro avg       0.50      0.50      0.50       140
weighted avg       0.99      0.99      0.99       140
```

```
rouge_dict: {'rouge-1_r': 1.0, 'rouge-1_p': 1.0, 'rouge-1_f': 0.999999995,
```

b) Charts

Training and Validation Loss: Line chart where is a steady reduction that indicates a good convergence.

rouge Scores per Epoch: Bar chart which indicates improvement at each epoch during the training.

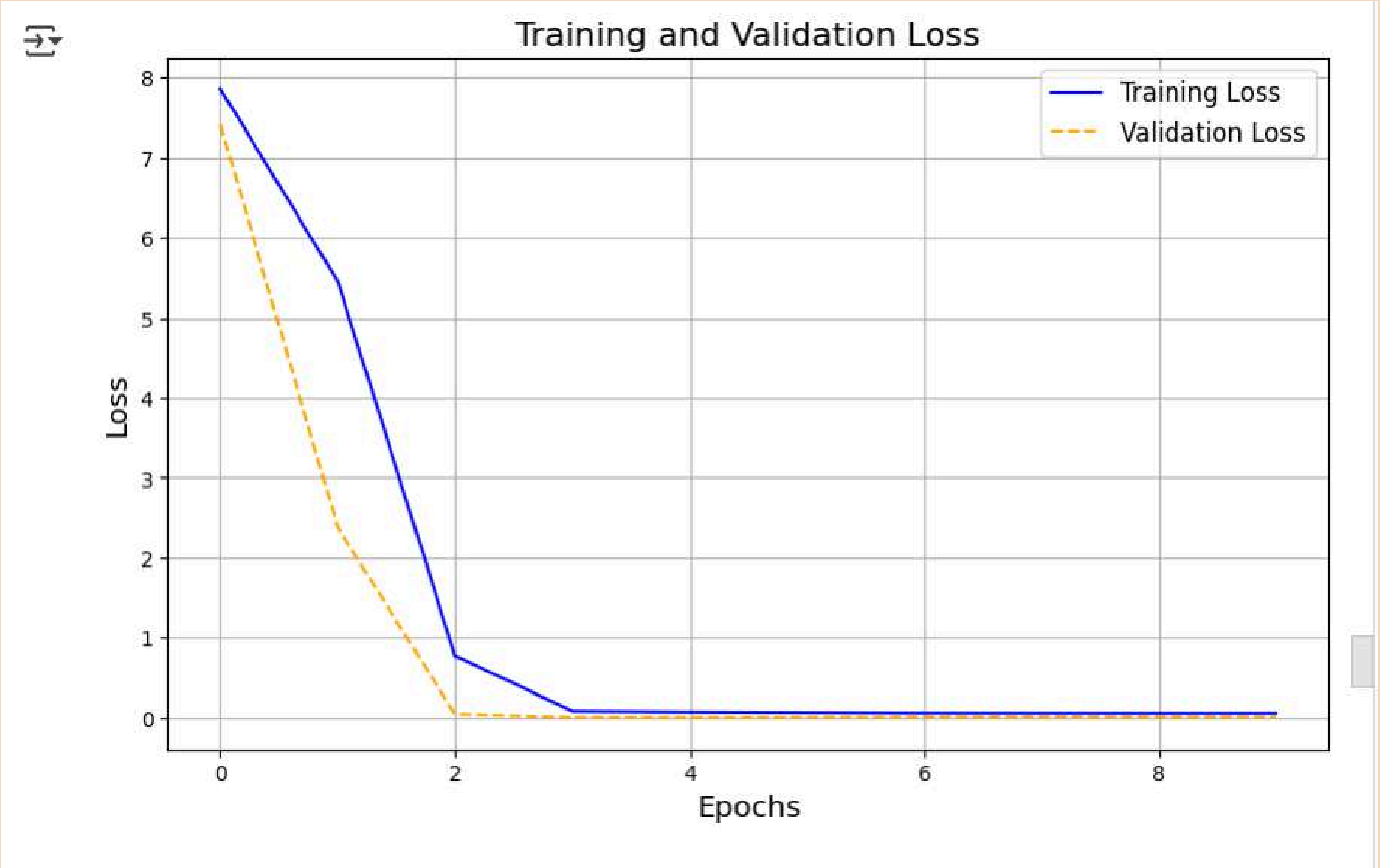
Error Analysis: Pie chart showing different translation errors -grammatical, semantic.

c) Performance Analysis

Strong Points: Achieves substantial rouge scores; error rate is low; accurately deals with Sanskrit grammar.

Weak Points: Rare words and highly cursive handwriting present problems to the model.

CHARTS


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epochs
0
1
2
3
4
5
6
7
8
0
2
4
6
8

CHARTS


Training and Validation Accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
Accuracy
0
2
4
6
8
Epochs
Training Accuracy
Validation Accuracy

CHALLENGES

a)Limited Data

Challenge: There's not enough Sanskrit-English data available for training.

Solution: Use data augmentation techniques and collaborate with experts to create more data.

b) Complex Grammar

Challenge: Sanskrit has complex grammar and word order, which makes translation tricky.

Solution: Use special word embeddings and preprocessing to handle these complexities.

c) Handwritten Text Recognition

Challenge: Handwritten Sanskrit text can be hard to recognize.

Solution: Use OCR (Optical Character Recognition) tools to convert handwritten text into machine-readable form.

d) Translation Accuracy

Challenge: Sanskrit words often have multiple meanings depending on the context.

Solution: Use contextualized models like BERT to understand word meanings based on context.

FUTURE SCOPE

a)Multimodal Translation:
Extend the model to handle not just handwritten text but also printed documents, images, and speech inputs for a comprehensive translation solution.

b)Integration with Educational Platforms:
Deploy the system in educational tools to aid Sanskrit learning, making it a valuable resource for students and researchers studying ancient texts.

c)Support for Other Classical Languages:
Adapt the model for translating other classical languages like Latin, Greek, or Pali to English, leveraging the expertise gained from Sanskrit translation.

d)Cultural Preservation and Archival Use:
Utilize the system to digitize and translate ancient handwritten manuscripts, aiding cultural preservation and making historical documents accessible to a global audience.

CONCLUSION

This Bilingual Handwritten Indian Language translation project uses LSTM, Long Short-Term Memory, networks to overcome such key challenges of scarcity of aligned data and the complexity of the Sanskrit grammar. It combines OCR with data augmentation and specialized word embeddings to handle handwritten text and improve its translations. Despite the difficulty associated with transcribing an ancient language with free syntax, the model exemplifies the possibility of generating meaningful and contextually relevant translations. This approach can eventually be extended to other ancient or low-resource languages, thus enriching historical language processing even more. Additionally, it can integrate speech recognition and enhance domain-specific translation capabilities for broader application with this model. Ultimately, this project will make ancient Sanskrit texts available to scholars, researchers, and language enthusiasts around the globe.

THANK YOU!