International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-10, Issue-01, Apr 2024

Multilingual Neural Machine Translation System for Indic Languages

[1]Prof. Vishal R. Shinde, [2]Mr. Pratham Agawane, [3]Miss. Nisha Sonawane, [4]Mr. Sanket Waghmare

[1]Asst.Professor,[2,3,4]UG Student,[1,2,3,4]Computer Engg. Dept. Shivajirao S. Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India. [1]mailme.vishalshinde@gmail.com, [2]pratham.agawane31@gmail.com, [3]nishasonavane0958@gmail.com, [4]sanketwaghmare508@gmail.com

Abstract - This research presents an LSTM-based language translation model utilizing the pre-trained Google/MT-small model within the transformer's architecture. The model is designed for bidirectional translations between various foreign languages and Indian languages. Three key features enhance its functionality: Text Translation, Image Translation, and Video Translation. To enhance user interaction, a user-friendly Flask API is developed, supporting all three translation features for seamless integration into applications or platforms. This research contributes to language translation and demonstrates practical applications in multilingual communication across text, image, and video modalities. The integration of transformers, EasyOCR, and MoviePy enhances adaptability, making the model valuable for diverse translation needs [1].

Keywords – Language Translation, Indian Language translation, LSTM, Transformers

I. INTRODUCTION

Language is the basis of human communication in recent times, but language conflicts can hinder intelligence and cooperation. Device Translation (MT) has become an effective tool to bridge these gaps, enabling seamless communication across multiple languages. This research project addresses the development of a unique translation model with the ability to translate between multiple languages with Indic languages, addressing the growing need for multilingual usability [1].

Today's world is interconnected and supports the requirement for impactful communication in many languages. But traditional machine translation techniques are generally legal, and it's a popular battle full of complex sentences, nuances, and conventions. New deep networks, especially short- term LSTM are specifically made to prevent long- term dependency problems. LSTM can analyze whole data streams, including data sequences, such as audio or video [13].

Transformer models have turned into the state's MT models and can last for a long time in a short time. Transformer was studied before using A substantial amount of new multilingual alphabets to analyze good representatives of modern languages. This pre- training reduces learning time and increases translation accuracy compared to teaching the model from scratch [11].

Google's MT-small Transformer Architecture is a pre-trained model designed for efficient machine translation [8]. This project extends its capabilities beyond text-to-text translation. This project introduces two new ideas:

Image Translation: This feature integrates the EasyOCR library for texts in images. The extracted text is then translated using this model and the translation is displayed, allowing users to easily translate the content of the text into an image.

Video Translation: MoviePy helps extract audio from videos. Google speaks and then converts the audio into text. This model translates this text and pyTTS converts the translated text back into audio. Finally, MoviePy seamlessly integrates translation into the video, ensuring good translation of video content.

II. AIMS AND OBJECTIVE

Aim:

The primary aim of the "Multilingual Neural Machine Translation System for Indic Languages" project is to develop a sophisticated and versatile language translation system that addresses the challenges posed by linguistic diversity, specifically focusing on Indic languages. It facilitates effective communication, promotes cultural exchange, and enhances accessibility to a variety of digital content

 DOI : 10.35291/2454-9150.2024.0269

Objective:

Leverage Pre-Trained Transformers: Employ Google's MT- small transformer architecture for efficient multi-lingual translation, including Indian languages.

Core Functionalities:

- **Text-to-Text Translation:** Seamlessly translate written content across different languages.
- **Image Translation** (using EasyOCR) Effortlessly translate text embedded in images.
- **Video Translation** (using MoviePy, Google Speech Recognition, pyTTS): Efficiently translate video content.

III. LITERATURE SURVEY

Paper 1: Machine translation using deep learning: An overview:

Deep Neural Networks (DNNs) are prominent in machine translation, with Recursive Recurrent Neural Networks (R2NNs) showing promise. R2NNs combine RNNs and recursive networks for reordering source and target language sentences. Word2vec generates word vectors and auto Encoder help to reconstruct them for the target language. However, RNN structures are complex to train, requiring powerful hardware like GPUs [7].

Paper 2: A Machine Translation System from Hindi to Sanskrit Language Using Rule-based Approach:

This paper proposes a rule-based machine translation system translating Hindi text to Sanskrit. It leverages the linguistic features of both languages to achieve the translation Evaluation of random sentences and tokens shows an accuracy of 94% for semantics and 86% for pragmatics. The distinctive linguistic characteristics of both languages are utilized for the translation process.

For system evaluation, two confusion matrices were generated, each comprising 50 randomly selected sentences and 100 tokens (words or phrases in Hindi) [2].

Paper 3: Deep Learning for Hindi Text Classification

This paper surveys various deep-learning architectures for text categorization tasks in this paper. The primary emphasis of the work lies in the classification of Hindi text. However, research on categorizing morphologically rich and low-resource Hindi written.

COMPARATIVE STUDY

Table No. 1 Comparative Anayalsis

SR NO.	PAPER TITLE	AUTHOR NAME	METHOD	PURPOSE
1.	Multilingual Neural Machine Translation System for Indic-to-Indic Languages	Das, S. B., Panda, D., Mishra, T. K., & Ekbal,	Build multilingual neural machine translation models	Improve machine translation for Indian languages
2.	Machine translation using deep learning: An overview	Shashi Pal Singh.	Recursive Recurrent Neural Networks (R2NNs)	Effective reordering for source-to-target languages
3.	A Machine Translation System from Hindi to Sanskrit Language using	Neha Bhadwal & Prateek Agrawal	Rule-based approach	High accuracy (94% for semantics, 86% for pragmatics)
4.	Deep Learning for Hindi Text Classification	Ramchandra Joshi.	Deep Learning architectures (CNNs, LSTMs, Transformers)	State-of-the-art results on NLP tasks

In the Devanagari script, has been limited due to not having a substantial labeled corpus. This work examines the deep learning architectures (CNNs, LSTMs, and Transformers) that are specifically utilized in Hindi text categorization problems. Research is hampered by the scarcity of large labeled Hindi corpora. The study compares pre-trained multilingual sentence embeddings like BERT and LASER 3 for text classification in Hindi and assesses models using translated English datasets [3].

IV. EXISTING SYSTEM

In existing systems, machine translation approaches are typically categorized into three main types: rule- based, statistical, and neural. Each of these approaches has its own strengths and limitations, shaping how effectively they can convert text between different languages.

Rule-based methods rely on predefined linguistic rules and patterns to perform translations. These systems excel in accuracy, especially within specific domains where language usage is consistent and predictable. For instance, in technical or scientific fields where language is highly structured, rule-based methods can produce translations with remarkable precision. However, their rigidity becomes apparent when faced with the fluidity and complexity of natural language outside their predefined rules. They often struggle to handle variations in context, leading to less fluent or inaccurate translations.

V. PROBLEM STATEMENT

The project aims to overcome the limitations of the currently in-use machine translation systems for Indic languages, which struggle with accuracy, context preservation, and handling the variety of languages spoken in the subcontinent of India. The goal is to develop a robust neural-machine

 DOI : 10.35291/2454-9150.2024.0269

International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-10, Issue-01, Apr 2024

translation system that can accurately and fluently translate between various Indic languages, overcoming these challenges and enabling more effective communication across linguistic boundaries.

PROPOSED SYSTEM

It has a multilingual stance and is built to support a variety of Indic language varieties. This approach offers several advantages, including the ability to translate between various language pairs without the need for separate language-specific models. Not only does this simplify model management, but it also proves beneficial for languages with limited linguistic resources and data.

This system proposes three features:

1. Text-to-text translation
2. Image Translation
3. Video Translation

Also, this model supports 5 Indian languages Hindi, Bengali, Tamil, Telugu, and Marathi. Five foreign languages are French, English, Russian, Japanese, and Spanish.

VI. ALGORITHM

The general idea of working of proposed system algorithms is given as follows:

Consensus Algorithm:

Step 1: Start

Step 2: Load the model

tokenizer=AutoTokenizer.from_pretrained(model_re po)

Step 3: Overview and quick test the model

Step 4: Test tokenizer

input_ids=tokenizer.encode(example_input _str,return_tensors='pt')

tokens=tokenizer.convert_ids_to_tokens(in put_ids[0)

sorted(tokenizer.vocab.items(), key=lambda x: x[1])

Step 5: Prepare Dataset

dataset = load_dataset('alt')

Step 6: Train the model

model.load_state_dict(torch.load(mode l_path)) optimizer = AdamW(model.parameters(), lr=lr)

Scheduler = get_linear_schedule_with_warmup(optimizer, n_warmup_steps, total_steps)

Step 7: Generate the Translation

TranslatedText=GetTranslationFromResult(T ransla tionResult)

Step 8: Improve Translation Quality if NecessaryIf the translation quality is still below the threshold:

If(TranslationQuality<ThresholdQuality):

ApplyPostProcessing(InputText, TranslatedText)

Step 9: Conclude the Translation Process

If there are more texts to translate, repeat the process from Step 1 for the next text

If there Are More Texts to Translate: Repeat Step1 for the next text Else:

End the translation process

EndTranslationProcess ()

Step 10: END.

VII. MATHEMATICAL MODEL

The mathematical model behind this system involves a combination of techniques:

1. Loss Functions

Loss functions like cross-entropy measure the quality of the translated text. BLEU score is a more complex metric that considers both fluency and accuracy. Customized functions can prioritize specific aspects of translation [4]

*Cross-entropy loss = -sum (p_i * log(q_i))*

2. Word Embeddings

In machine translation, word embedding plays a crucial role. These are word vector representations that have been taught and capture semantic links. Words with similar meanings (e.g., "**king**" and "**queen**") will possess vector representations that are comparable, while words with different meanings (e.g., "**king**" and "**car**") will have distinct vectors. Techniques like Word2Vec or Fast Text learn these embeddings from large text corpora. Word embedding is then utilized in various tasks, including machine translation [10].

3. Optimization

Machine translation models rely on optimization algorithms to learn effectively. These algorithms adjust the model's internal factors to lessen the loss function's magnitude, ultimately improving translation quality. Two popular choices in machine translation are Adam and SGD [9].

3.1 Adam (Adaptive Moment Estimation):

This is a sophisticated optimizer that adapts the learning rate for each parameter during training. It's particularly well-suited for training neural networks used in machine translation. This algorithm is used in machine translation as Adam. Adam is a stochastic gradient descent optimizer that is adaptive to the learning rate. The Adam update rule is defined as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$
$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot m_t}{\sqrt{v_t} + \epsilon}$$

 DOI : 10.35291/2454-9150.2024.0269

where:

m_t is the first moment estimate of the gradient v_t is the second moment estimate of the gradient θ_t are the model parameters α is the learning rate β_1 and β_2 are hyper parameters that control the decay of the first and second-moment estimates, respectively ε is a small value to prevent division by zero.

3.2 SGD (Stochastic Gradient Descent):

A simpler yet fundamental optimizer, SGD is often used as a baseline in machine learning research. It iteratively adjusts the model parameters according to the calculated gradients.

4. Transformers

The prevailing method in machine translation is now using transformers, a potent neural network design. Transformers do exceptionally well at identifying long-range dependencies inside words, which is essential for precise translation, in contrast to conventional to the recurrent neural networks (RNNs). The model is able to attend to pertinent portions of the original phrase when producing the target translation, thanks to a self-attention mechanism. [6]

4.1 Scaled Dot-Product Attention:

This forms the essential foundation of self-attention. It computes attention scores between every word within the input list, signifying the relevance of each word to the current word under processing. The formula for scaled dot-product attention is:

Attention (Q, K, V) = soft max ((Q * K^T) / √dk) * V

Where:

Q (Query): A vector representing the current word being processed.

K (Key): A vector representing all words in the input Sequence.

V (Value): A vector containing information about all words within the input line.

dk: The dimension of the key and value vectors

VIII. SYSTEM ARCHITECTURE

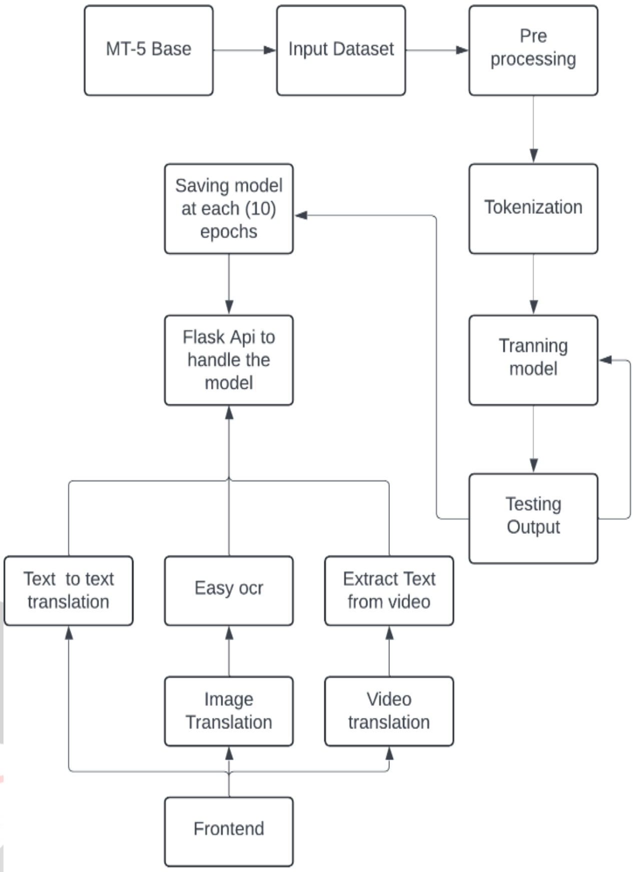

Fig 1. System Architecture

Description:

Recurrent neural network (RNN) design includes Long Short-Term Memory (LSTM) networks, which are renowned for their capacity to efficiently recognize long-term relationships in data that are sequential. Within the framework of translation tasks, LSTMs are utilized within an encoder- decoder framework, a popular architecture for sequence-to-sequence learning. LSTMs in translation use an encoder-decoder architecture. The encoder (LSTM) processes the source sentence, capturing its meaning. The decoder (another LSTM) leverages this information to produce the intended translation word by word. [12]

IX. ADVANTAGES

Multilingual Translation: Supports translation between five Indian languages, facilitating communication and understanding among diverse linguistic backgrounds.

Image-to-Text Translation: This process extracts text from images and then interprets it in many languages, thereby enhancing accessibility and understanding of visual content.

Text-to-Text Translation: Translates documents, emails,

 DOI : 10.35291/2454-9150.2024.0269

ISSN : 2454-9150 Vol-10, Issue-01, Apr 2024

websites, and textual content between languages, enabling efficient communication across linguistic borders. Comprehensibility of multimedia content

Video Translation: clear comprehension and accessibility across language barriers by transcribing and translating video material.

XII. DESIGN DETAILS

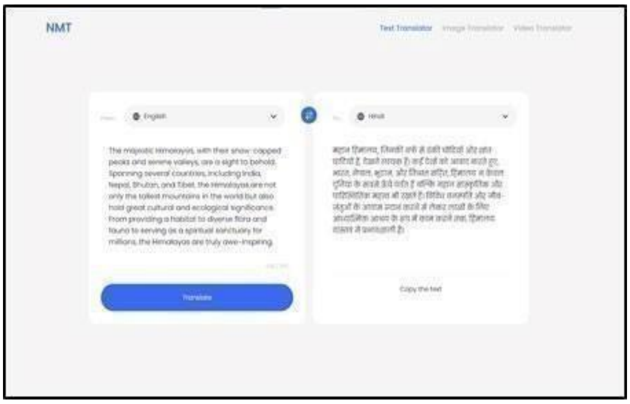

Fig 2. Text-to-text translation

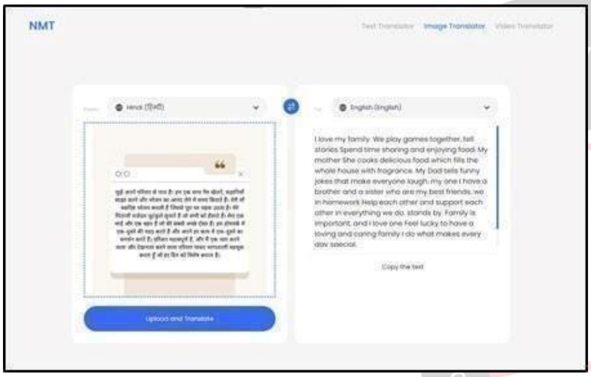

Fig 3. Image-to-text translation

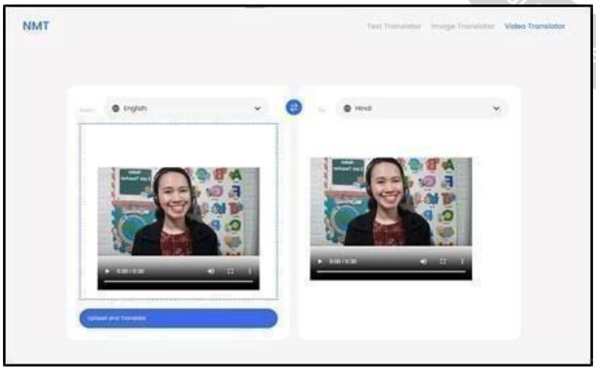

Fig 4. Video to Video translation

XIII. CONCLUSION

Thus, we have tried to implement the paper **"Sudhansu Bala Das, Divya Jyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, Asif Ekbal" (2023) "Multilingual Neural Machine Translation System for Indic-to-Indic Languages"**, into practice and according to the implementation the conclusion is as follows, this model marks a significant advancement in getting beyond linguistic obstacles and promoting communication across diverse linguistic contexts. By harnessing sophisticated techniques such as LSTM architecture and integrating the MT-5 Base pre-trained dataset library, this project has successfully crafted a robust and adaptable translation system. This system encompasses a rich array of features, including text-to-text translation, image translation, and video translation, all finely tuned to accommodate the complexities inherent in Indic languages and beyond. Testing has confirmed the effectiveness and accuracy of translation algorithms in different scenarios, instilling confidence in the system's reliability.

REFERENCES

[1] Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, Asif Ekbal "Multilingual Neural Machine Translation System for Indic-to-Indic Languages" (2023)

[2] Neha Bhadwal, Prateek Agrawal, Vishu Madaan "A Machine Translation System from Hindi to Sanskrit Language using Rule-based Approach" (2020)

[3] Ramchandra Joshi, Purvi Goel, Raviraj Bhuminand Joshi "Deep Learning for Hindi Text Classification" (2020)

[4] Jurafsky Daniel, and James H. Martin. "Speech and Language Processing" (3rd Edition). Pearson, (2019).

[5] Lample, G., Conneau, A., Denil, M., Joulin, A., Membrado, J., Roux, N., Uszkoreit, J. "Cross-lingual language modeling." (2019).

[6] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 31 (2017).

[7] Shashi Pal Singh, Ajai Kumar, Hemant Darbari "Machine translation using deep learning: An overview" (2017).

[8] Transformer: A Novel Neural Network Architecture for Language Understanding. (2022)

[9] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization (2014).

[10] Sutskever, I., Vinyals, O., & Le, Q. V. Sequence to sequence learning with neural networks. (2014)

[11] Mikolov, Tomas, et al. "Distributed Representations of words and phrases andtheir compositionality." Advances in neural information processing systems. (2013).

[12] Hochreiter, S., & Schmidhuber, J. Long short-term memory. Neural computation. Research gate. (1997).

[13] Prof. Kanchan Umavane, Avhad, P., Tavanoji, R., & Pardeshi, R. LSTM based Malware Detection Framework for Android Devices IJREAM, ISSN: 2454-9150, Volume 09, Issue 01, APR 2023 Special Issue

 DOI : 10.35291/2454-9150.2024.0269