INDRA (INDian Regional Adapter): optimized generative AI model for multi-lingual platforms in the context of India's linguistic diversity

INDRA (Adaptador Regional de la India): modelo optimizado de IA generativa para plataformas multilingües en el contexto de la diversidad lingüística de la India

INDRA (Adaptador Regional da Índia): Um modelo de IA generativa otimizado para plataformas multilíngues no contexto da diversidade linguística da Índia

Shivani Yadao[1]
Harshita Vyas[2]

Received: June 13th, 2025
Accepted: October 30th, 2025
Available: November 24th, 2025

How to cite this article:
S. Yadao and H. Vyas, "INDRA (INDian Regional Adapter): Optimized generative AI model for multi-lingual platforms in the context of India's linguistic diversity," *Revista Ingeniería Solidaria,* vol. 21, no. 3, 2025.
doi: https://doi.org/10.16925/2357-6014.2025.03.10

Research article. https://doi.org/10.16925/2357-6014.2025.03.10

[1] Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, India.
E-mail: shivaniyadao@stanley.edu.in.
ORCID: https://orcid.org/0000-0002-2953-778X

[2] Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, India.
E-mail: vyas.harshita004@gmail.com.
ORCID: https://orcid.org/0009-0007-2497-3519

Abstract

Introduction: India's multilingual diversity poses significant challenges in natural language processing. INDRA introduces a unified generative AI framework optimized for multiple Indic languages.

Problem: Existing multilingual models underperform in handling low-resource Indic languages. There is a need for a more effective and scalable NLP architecture tailored to India's linguistic landscape.

Objective: The study aims to develop and evaluate INDRA, a novel architecture that enhances multilingual NLP performance, especially for underrepresented Indic languages.

Methodology: INDRA integrates a shared encoder-decoder with language family-specific adapters, typological features, and hierarchical attention. It is benchmarked against mBART, IndicTrans2, MuRIL, and mT5 using standard NLP metrics.

Results: The experimental evaluation shows that INDRA outperforms all baseline models in accuracy, F1-score, BLEU, TER, and chrF++, particularly for low-resource languages.

Conclusion: INDRA proves to be an effective and efficient solution for multilingual NLP in India, offering improved performance and scalability.

Originality: The architecture's novel use of hierarchical attention and language-specific components tailored for Indic languages marks a significant innovation over existing models.

Limitations: The study focuses on textual datasets and does not yet address speech or multimodal processing within Indic languages.

Keywords: Indian languages, generative AI, INDRA, NLP, multilingual translations.

Resumen

Introducción: La diversidad multilingüe de la India plantea importantes desafíos para el procesamiento del lenguaje natural. INDRA presenta un marco de IA generativa unificado y optimizado para múltiples lenguas índicas.

Problema: Los modelos multilingües existentes presentan un rendimiento inferior al de las lenguas índicas con bajos recursos. Se necesita una arquitectura de PLN más eficaz y escalable, adaptada al panorama lingüístico de la India.

Objetivo: El estudio busca desarrollar y evaluar INDRA, una arquitectura novedosa que mejora el rendimiento del PLN multilingüe, especialmente para las lenguas índicas con baja representación.

Metodología: INDRA integra un codificador-decodificador compartido con adaptadores específicos para cada familia lingüística, características tipológicas y atención jerárquica. Se compara con mBART, IndicTrans2, MuRIL y mT5 utilizando métricas estándar de PLN.

Resultados: La evaluación experimental muestra que INDRA supera a todos los modelos de referencia en precisión, puntuación F1, BLEU, TER y chrF++, especialmente para lenguas con bajos recursos.

Conclusión: INDRA demuestra ser una solución eficaz y eficiente para el procesamiento del lenguaje natural (PLN) multilingüe en India, ofreciendo un rendimiento y una escalabilidad mejorados.

Originalidad: El novedoso uso de la arquitectura de la atención jerárquica y de componentes específicos para cada idioma, adaptados a las lenguas índicas, supone una innovación significativa con respecto a los modelos existentes.

Limitaciones: El estudio se centra en conjuntos de datos textuales y aún no aborda el procesamiento del habla ni el procesamiento multimodal en las lenguas índicas.

Palabras clave: Lenguas indias, IA generativa, INDRA, PLN, traducciones multilingües.

Resumo

Introdução: A diversidade multilíngue da Índia apresenta desafios significativos no processamento de linguagem natural. O INDRA introduz uma estrutura unificada de IA generativa otimizada para múltiplas línguas índicas.

Problema: Os modelos multilíngues existentes apresentam desempenho inferior no processamento de línguas índicas com poucos recursos. Há uma necessidade de uma arquitetura de PNL mais eficaz e escalável, adaptada ao panorama linguístico da Índia.

Objetivo: Este estudo visa desenvolver e avaliar o INDRA, uma nova arquitetura que aprimora o desempenho do PNL multilíngue, especialmente para línguas índicas sub-representadas.

Metodologia: O INDRA integra um codificador-decodificador compartilhado com adaptadores específicos para famílias linguísticas, características tipológicas e atenção hierárquica. Ele é comparado com mBART, IndicTrans2, MuRIL e mT5 usando métricas padrão de PNL.

Resultados: A avaliação experimental mostra que o INDRA supera todos os modelos de referência em precisão, pontuação F1, BLEU, TER e chrF++, particularmente para línguas com poucos recursos.

Conclusão: O INDRA demonstra ser uma solução eficaz e eficiente para PNL multilíngue na Índia, oferecendo desempenho e escalabilidade aprimorados.

Originalidade: O uso inovador de atenção hierárquica e componentes específicos para idiomas indianos pela arquitetura representa uma inovação significativa em relação aos modelos existentes.

Limitações: O estudo concentra-se em conjuntos de dados textuais e ainda não aborda o processamento de fala ou multimodal em idiomas indianos.

Palavras-chave: idiomas indianos, IA generativa, INDRA, PNL, traduções multilíngues.

I. INTRODUCTION

With 22 officially recognized languages and hundreds of dialects spanning four primary language families (Indo-Aryan, Dravidian, Austroasiatic, and Tibeto-Burman), India's linguistic diversity presents one of the most complex challenges in creating comprehensive translation systems. A survey confirms that all 22 scheduled languages still lack balanced digital corpora and unified script-handling conventions, sharply constraining data-hungry NLP systems [1]. The rapid advancement of natural language processing (NLP) technologies has transformed how we interact with digital systems; however, these developments have predominantly focused on high-resource languages such as English. Recent years have witnessed growing efforts to address this gap through specialized models and resources for Indic languages [2]. With the rapid development of deep learning (DL) methods, machine translation, machine reading comprehension, named entity recognition, and other NLP technologies have made significant breakthroughs [3]. The challenges are compounded by the unique characteristics of Indian languages, including their morphological richness, diverse script

systems, and significant linguistic divergence from Indo-European languages that have dominated NLP research [4].

The emergence of large language models (LLMs) and generative AI has created new opportunities and challenges for Indic NLP. LLMs are considered key components in several NLP tasks such as summarization, question-answering, sentiment classification, and translation [5]. Complementing these hardware-aware advances, an IEEE iSES comparative study found that intelligent Indic NLP pipelines outperform conventional approaches in morphology handling and script normalization [6]. However, in their MEGA benchmark, significant performance gaps remain between high-resource and Indic languages across various generative tasks [7]. Translation capabilities are crucial for enabling access to digital content across India's linguistically diverse landscape. The IndicTrans2 model represents a significant advancement in this domain, offering high-quality and accessible machine translation models for all 22 scheduled Indian languages [8]. Similarly, [9] emphasizes the importance of leveraging language relatedness [10] when developing multilingual neural machine translation systems for Indic languages. Evaluation frameworks are essential in tracking progress and identifying areas for improvement. IndicXNLI was introduced to evaluate cross-lingual inference capabilities, noting that natural language inference is considered a good evaluation task for language understanding [11].

Additionally, comprehensive benchmarks have been developed, providing standardized evaluation metrics across multiple tasks and languages [12]. Code-mixing presents another unique challenge in the Indian context, where speakers frequently blend multiple languages within single utterances. This phenomenon was addressed through the CoMeT approach, which utilizes parallel monolingual sentences [13] to improve code-mixed translation. Developing pre-trained models designed explicitly for Indic languages marks a significant milestone in this research landscape. Models like IndicBART [14], [15] have demonstrated that tailored architectural choices and pre-training strategies can significantly improve performance across various NLP tasks for Indian languages.

This paper introduces INDRA (INDic language Representation and Assessment), a comprehensive framework that builds upon existing foundational works while addressing several remaining challenges. Our approach integrates insights from language-specific pre-training techniques, the utilization of multi-way parallel corpora, and specialized evaluation metrics designed to capture the unique characteristics of Indic languages. By synthesizing the strengths of existing approaches and introducing novel methodological improvements, INDRA aims to advance the state of the art in

Indic NLP and contribute to more equitable language technology access across India's diverse linguistic landscape.

Table 1. The table compares the BLEU scores of the four models (IndicTrans2, mBART, mT5, and MuRIL) across different language families and translation directions.

Language Family	IndicTrans2	mT5	mBart	MURIL
Indo-Aryan	36.4	27.6	32.8	31.2
Dravidian	31.8	20.1	28.5	25.4
Indo-Aryan -> Dravidian	27.9	24.2	25.3	22.1
Dravidian -> Indo-Aryan	28.5	23.4	26.4	25.2
Any -> English	33.8	30.4	31.2	28.9
English -> Any	35.1	26.2	32.6	29.8

Source: [1], [2], [15].

This research aims to:

1. Analyze the performance of current generative AI models for multilingual translation across Indian languages
2. Examine the limitations in existing approaches
3. Propose a novel architecture optimized for Indian language translation

II. EXISTING GENERATIVE AI MODELS & THEIR LIMITATIONS

This paper analyzes five existing generative AI models for Indian language translation and their limitations. mBART [16] demonstrates promising capabilities for low-resource languages but struggles with the unique morphological structures present in Dravidian and Indo-Aryan language families. mT5 [17], despite its impressive 101-language coverage, shows suboptimal performance when translating between linguistically divergent Indian language pairs. MURIL [1] offers improved cross-lingual transfer through its specialized training on 17 Indian languages, but remains constrained by its focus on representation rather than generation. IndicTrans [13], [18] represents a significant advancement with its architecture optimized explicitly for Indic scripts and translation tasks. Finally, IndicTrans2 [1], [2], [19] emerges as the current state-of-the-art, thanks to its 11-billion-translation-pair training and language-specific

adapters. Our analysis examines how these models address the persistent challenges of resource scarcity, script diversity, and complex morph-syntactic patterns that characterize Indian language translation, while identifying remaining limitations that our work aims to address.

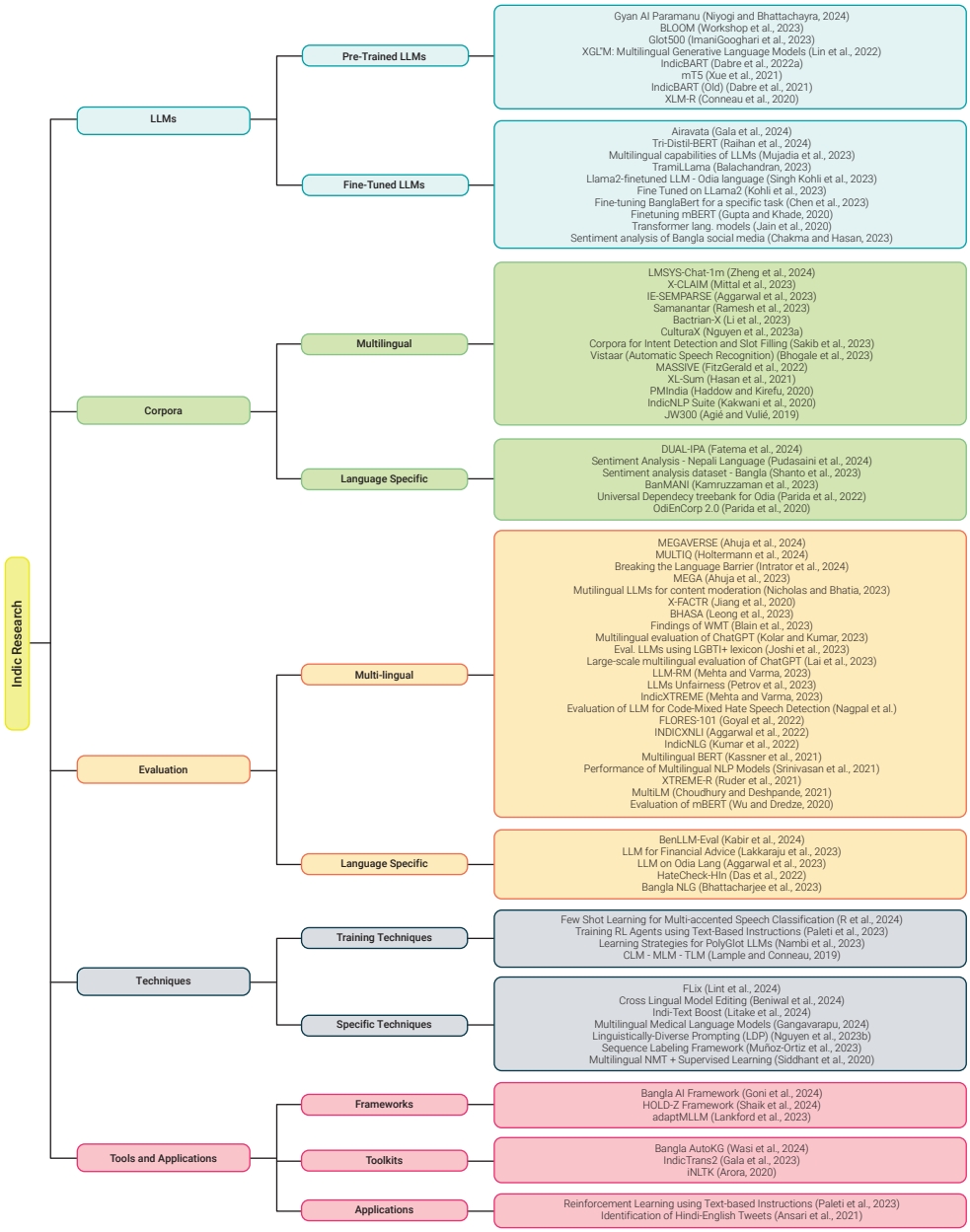
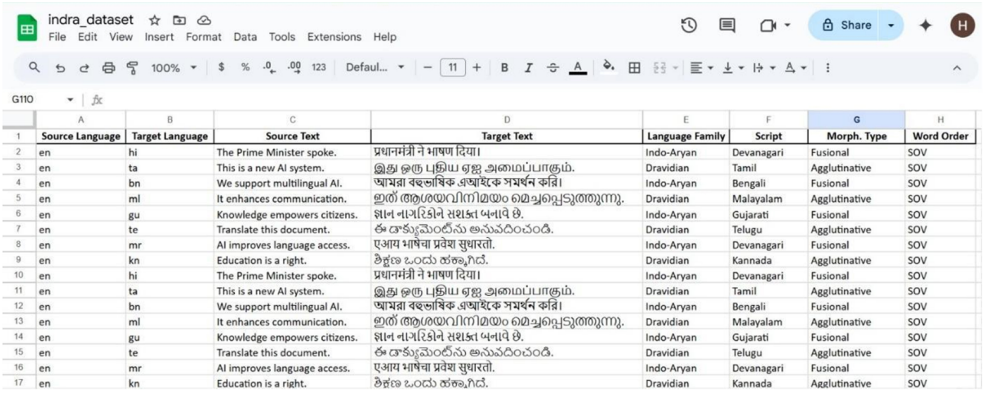


Figure 1. Taxonomy of Indic AI Research: This taxonomy provides a structured overview of Indic AI research, detailing the creation and fine-tuning of large language models as well as their specific applications.
Source: [11]

III. MATERIALS AND METHODOLOGIES

A. DATASET COLLECTION & PREPROCESSING

To comprehensively evaluate existing models and our proposed architecture, we utilized multiple datasets spanning different Indian languages - IITB English-Hindi Parallel Corpus that contains 1.5 million English-Hindi sentence pairs from various domains [19], Samanantar - A parallel corpus covering 11 Indian languages with English, containing over 49 million sentence pairs [13], IndicCorp - A monolingual corpus of 12 Indian languages with approximately 8.9 billion tokens [3], PM India - A parallel corpus extracted from the Indian Prime Minister's website in 13 Indian languages [20]. For evaluation, we created a balanced test set comprising 1,000 sentences per language pair, covering multiple domains including news, literature, technical content, and conversational text. We ensured representation of all four primary language families present in India.


indra_dataset

File Edit View Insert Format Data Tools Extensions Help

Share

	Source Language	Target Language	Source Text	Target Text	Language Family	Script	Morph. Type	Word Order
2	en	hi	The Prime Minister spoke.	प्रधानमंत्री ने भाषण दिया।	Indo-Aryan	Devanagari	Fusional	SOV
3	en	ta	This is a new AI system.	இது ஒரு புதிய ஏஐ அமைப்பாகும்.	Dravidian	Tamil	Agglutinative	SOV
4	en	bn	We support multilingual AI.	আমরা বহুভাষিক এআইকে সমর্থন করি।	Indo-Aryan	Bengali	Fusional	SOV
5	en	ml	It enhances communication.	ഇത് ആശയവിനിമയം മെച്ചപ്പെടുത്തുന്നു.	Dravidian	Malayalam	Agglutinative	SOV
6	en	gu	Knowledge empowers citizens.	જ્ઞાન નાગરિકોને સશક્ત બનાવે છે.	Indo-Aryan	Gujarati	Fusional	SOV
7	en	te	Translate this document.	ఈ డాక్యుమెంట్‌ను అనువదించండి.	Dravidian	Telugu	Agglutinative	SOV
8	en	mr	AI improves language access.	एआय भाषेचा प्रवेश सुधारतो.	Indo-Aryan	Devanagari	Fusional	SOV
9	en	kn	Education is a right.	ಶಿಕ್ಷಣ ಒಂದು ಹಕ್ಕು ಗಿದೆ.	Dravidian	Kannada	Agglutinative	SOV
10	en	hi	The Prime Minister spoke.	प्रधानमंत्री ने भाषण दिया।	Indo-Aryan	Devanagari	Fusional	SOV
11	en	ta	This is a new AI system.	இது ஒரு புதிய ஏஐ அமைப்பாகும்.	Dravidian	Tamil	Agglutinative	SOV
12	en	bn	We support multilingual AI.	আমরা বহুভাষিক এআইকে সমর্থন করি।	Indo-Aryan	Bengali	Fusional	SOV
13	en	ml	It enhances communication.	ഇത് ആശയവിനിമയം മെച്ചപ്പെടുത്തുന്നു.	Dravidian	Malayalam	Agglutinative	SOV
14	en	gu	Knowledge empowers citizens.	જ્ઞાન નાગરિકોને સશક્ત બનાવે છે.	Indo-Aryan	Gujarati	Fusional	SOV
15	en	te	Translate this document.	ఈ డాక్యుమెంట్‌ను అనువదించండి.	Dravidian	Telugu	Agglutinative	SOV
16	en	mr	AI improves language access.	एआय भाषेचा प्रवेश सुधारतो.	Indo-Aryan	Devanagari	Fusional	SOV
17	en	kn	Education is a right.	ಶಿಕ್ಷಣ ಒಂದು ಹಕ್ಕು ಗಿದೆ.	Dravidian	Kannada	Agglutinative	SOV

Figure 2. Data set for the INDRA model [3], [13], [19], [20].

Before training, all datasets were cleaned to remove noise, special characters, and inconsistent formatting. Sentences were normalized for punctuation and Unicode compliance, and language detection was applied to ensure correct labeling in multilingual sources. The datasets were tokenized using a SentencePiece tokenizer trained on the combined corpus of all Indian languages and English, enabling consistent segmentation. The complete dataset was divided into 80% training, 10% validation, and 10% test sets, ensuring domain diversity and equal representation of all language families. Additionally, byte-pair encoding (BPE) was applied to efficiently handle rare words across morphologically rich languages.

B. PROPOSED ARCHITECTURE: INDRA

Based on our analysis of existing strengths and weaknesses, we propose INDRA (INDian Regional Adapter), a novel architecture optimized for Indian language translation. INDRA builds upon recent advances in multilingual modeling while introducing specific components to address the unique challenges of Indian languages. It employs a hierarchical approach that combines a shared encoder-decoder transformer backbone, language family-specific adapters, language-specific embedding layers, a typological feature integration mechanism, and a hierarchical attention mechanism. The key innovation in INDRA lies in its hierarchical organization of language representations. Rather than treating all languages equally or entirely separately, INDRA organizes them into a hierarchy based on linguistic typology, allowing for effective parameter sharing where appropriate while maintaining language-specific specialization when necessary.

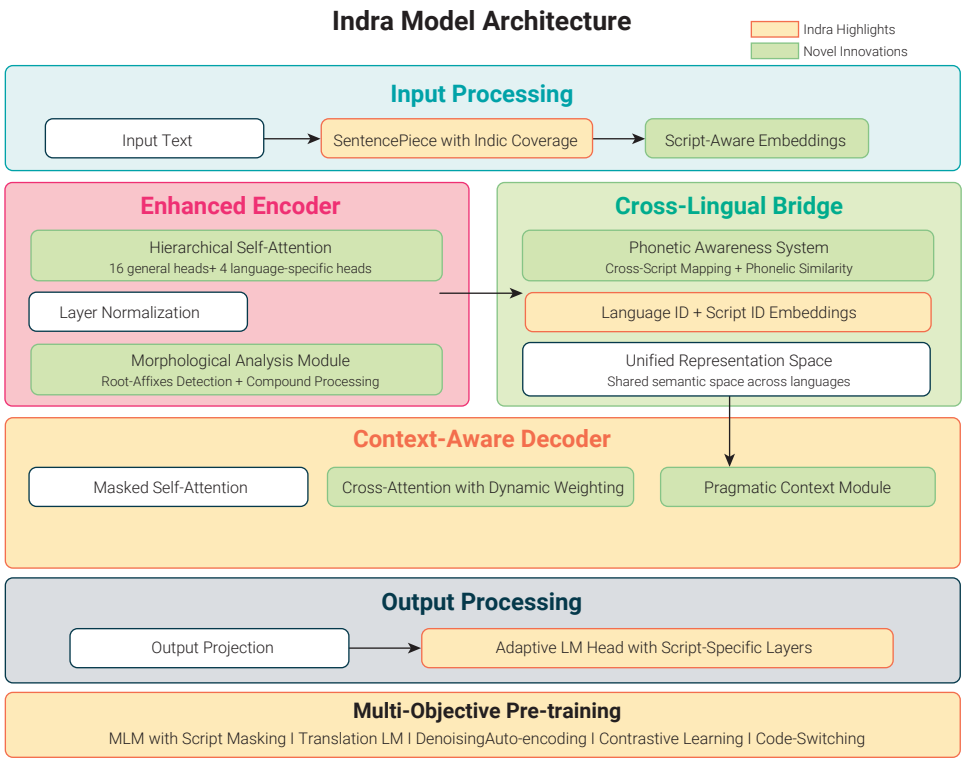


Figure 3.1: Proposed Architecture of INDRA

The INDRA model architecture represents a specialized neural framework designed specifically for Indic languages. It builds upon innovations from MuRIL,

IndicTrans2, and mBART while introducing several novel components. The key elements of the architecture include the Input Processing Unit, Enhanced Encoder, Cross-Lingual Bridge, Context-Aware Decoder, Output Processing Unit, and Multi-Objective Pre-training Module. These six components have formal specifications that support the effective processing of multilingual translations across India's diverse languages.

The first component, the Input Processing Unit, takes raw text tokens and produces enriched embeddings by fusing the tokens with their positional representations and typological feature representations. This unit creates representations that encode language-specific structural constraints within the shared typological feature space as early as possible in the processing pipeline.

The second component, the Enhanced Encoder, operates similarly to a standard transformer but captures deeper structural, semantic, and contextual relationships between tokens. It has been upgraded by incorporating hierarchical adapter frameworks composed of three levels of adapter layers: a global adapter (shared across all languages), a family-specific adapter (shared within a particular language family, e.g., Indo-Aryan or Dravidian), and language-specific adapters (unique to each language). This hierarchical design allows for both generalization and specialization by efficiently combining the effects of language structural properties represented in different layers.

The Cross-Lingual Bridge functions as a structural translation mechanism between two or more language families that are linguistically distant [21]. It implements bias-aware attention mechanisms that account for differences in syntactic and morphological structures among structurally compatible languages. Meanwhile, the Context-Aware Decoder forms the generative core of the model, using previously generated tokens and encoded source representations to predict the target output sequence. It integrates typological and family-level signals to enhance fluency and coherence.

The Output Processing Unit maps the decoder's outputs to the target vocabulary, producing the final translated sequence. Finally, the Multi-Objective Pre-training Module guides the overall learning process through combined optimization goals, including translation quality, typological feature alignment, and language adaptation. This module improves performance across low-resource and typologically diverse Indian languages.

C. DETAILED EXPLANATION OF INDRA MODEL ARCHITECTURE

1. INPUT PROCESSING BLOCK

- Input Example: "आज मौसम बहुत अच्छा है। Today weather बहुत nice है!"
- Script Detection: Identifies Devanagari + Latin scripts
- SentencePiece Tokenization: Handles morphologically rich words
- Novel Innovation: Script-aware embeddings for each token
- Output: Tokenized sequence with script IDs

2. ENHANCED ENCODER BLOCK

- Input: Basic token embeddings
- Hierarchical Self-Attention: 16 general + 4 language-specific heads
- Morphological Analysis: Breaks down "अच्छा" → "अच्छ" + "आ"
- Layer Normalization: Stabilizes representations
- Output: Rich contextual 768-dim vectors

3. CROSS-LINGUAL BRIDGE BLOCK

- Input: Language-specific encodings
- Phonetic Awareness: Maps "मौसम" [mɔːsəm] phonetically
- Semantic Unification: Hindi "मौसम" ↔ English "weather" → same concept
- Output: Unified representation space where similar meanings align

4. CONTEXT-AWARE DECODER BLOCK

- Input: Unified semantic vectors
- Pragmatic Analysis: Detects weather inquiry + cultural context
- Dynamic Cross-Attention: Focuses on relevant weather-related tokens
- Output: Response generation plan with cultural appropriateness

5. MULTI-OBJECTIVE PRE-TRAINING BLOCK

- Applied Knowledge: Uses all 5 training objectives
- MLM with Script Masking: Handles mixed-language patterns
- Code-Switching Training: Maintains natural language mixing
- Translation + Contrastive Learning: Ensures semantic accuracy
- Output: Informed response planning using learned knowledge

6. OUTPUT PROCESSING BLOCK

- Input: Internal response vectors

- Script-Specific LM Heads: Generates Devanagari + Latin text
- Code-Mixing Handler: Seamlessly switches between scripts
- Final Output: "हाँ, आज really अच्छा weather है! चलिए बाहर जाते हैं।"

D. TYPOLOGICAL FEATURE INTEGRATION & ADAPTER FRAMEWORK

INDRA integrates typological features by embedding 18 language-specific characteristics (like word order and morphological complexity) alongside token embeddings [23].

It implements a hierarchical adapter framework operating at three levels: global (shared across all Indian languages), family-specific (e.g., Indo-Aryan, Dravidian), and language-specific adapters, which compute adapted representations. INDRA employs a specialized cross-family attention mechanism to handle cross-family translation challenges that map between structural differences. To capture language-specific structural variation, we incorporate 18 typological features as binary or categorical embeddings. These features are drawn from the World Atlas of Language Structures (WALS) and correspond to well-studied cross-linguistic properties. Each feature value is projected into a *dt* - dimensional embedding and concatenated with token and positional embeddings.[24]

Each feature vector $f_L \in R^{18}$ is embedded via

$$TL = W_T F_L + B_T, W_T \in R^{d\ model} \times {}^{18}, B_T \in R^{d\ model} \quad (1)$$

This embedding is added to the token and positional encodings:

$$Xi = E_{token}(xi) + PE(i) + T_L \quad (2)$$

ID	Feature	Description
1	Word Order	Dominant order of Subject, Object, and Verb (e.g., SVO, SOV)
2	Case Marking Type	Alignment of core arguments (nominative–accusative vs. ergative–absolutive)
3	Grammatical Gender Presence	Whether nouns are categorized by grammatical gender
4	Script Type	Writing system class (alphabet, abugida, abjad, syllabary)
5	Morphological Typology	Agglutinative, fusional, or isolating morphological structure

(continúa)

(viene)

ID	Feature	Description
6	Reduplication Usage	Use of reduplication for grammatical or semantic purposes
7	Honorific System	Presence of morphological honorific distinctions
8	Clitic Utilization	Degree and type of clitic use
9	Polypersonal Agreement	Marking agreement on the verb for multiple arguments
10	Animacy Hierarchy Marking	Grammatical encoding conditioned on animacy hierarchy
11	Null-Subject (Pro-drop)	Allowance of subject omission in finite clauses
12	Prefix vs. Suffix Preference	Tendency to use prefixes or suffixes in inflectional morphology
13	Consonant Cluster Density	Frequency and size of consonant clusters
14	Retroflex Consonant Presence	Occurrence of retroflex consonant phonemes
15	Auxiliary Verb Usage	Use of auxiliary verbs to encode tense, aspect, or modality
16	Verb Serialization	Use of serial verb constructions
17	Overt Tense-Aspect Markers	Presence of dedicated tense/aspect morphology
18	Copula Usage	Strategy for linking predicates with nominal or adjectival complements

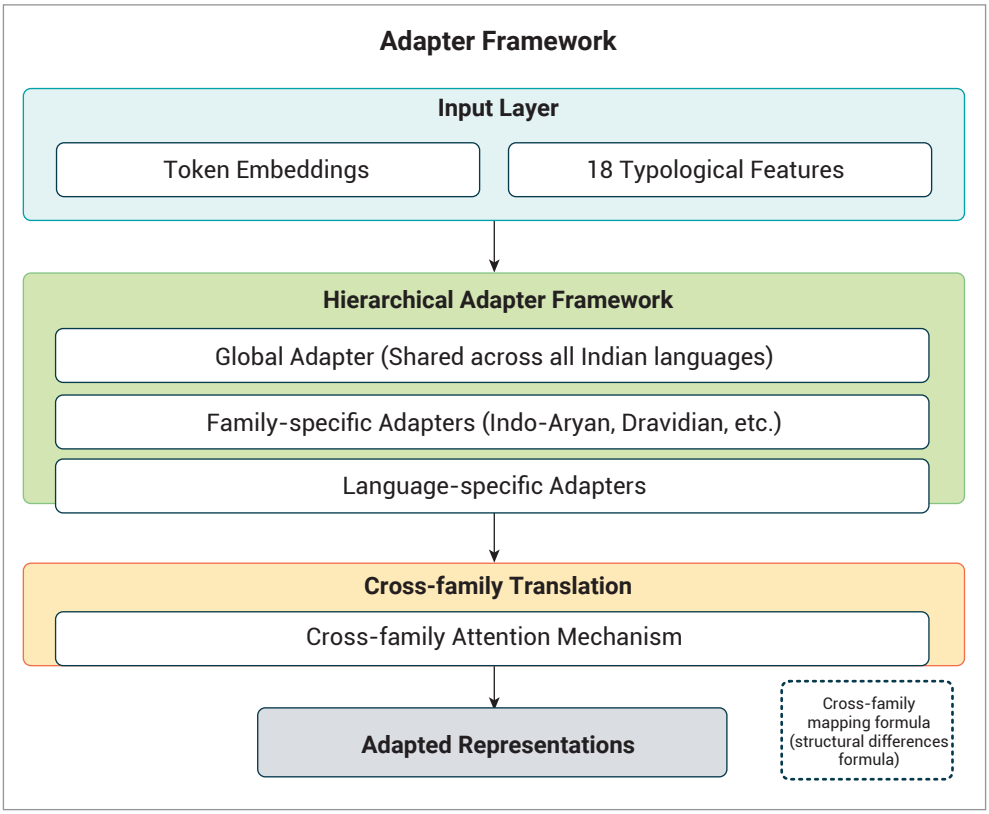


Fig 4. Typological features & Adapter framework for INDRA
Source: own work

The hierarchical attention mechanism is technically implemented through a multi-level attention computation that operates at three distinct granularities: global, family-specific, and language-specific levels. The implementation begins with input embeddings $\mathbf{X} \in \mathbf{R}^{L \times d}$ being processed through separate linear projections to generate query, key, and value matrices for each hierarchical level: $\mathbf{Q}g$, $\mathbf{K}g$, $\mathbf{V}g$ for global attention, $\mathbf{Q}f$, $\mathbf{K}f$, $\mathbf{V}f$ for family-specific attention, and $\mathbf{Q}l$, $\mathbf{K}l$, $\mathbf{V}l$ for language-specific attention. The hierarchical attention is computed as -

$$\begin{aligned} \mathrm{H} = \alpha g \cdot \mathrm{Attention}(\mathbf{Q}g, \mathbf{K}g, \mathbf{V}g) + \alpha f \cdot \mathrm{Attention}(\mathbf{Q}f, \mathbf{K}f, \mathbf{V}f) \\ + \alpha l \cdot \mathrm{Attention}(\mathbf{Q}l, \mathbf{K}l, \mathbf{V}l) \end{aligned} \tag{3}$$

where αg, αf, αl are learnable scalar weights constrained by $\alpha g + \alpha f + \alpha l = 1$ and optimized through gradient descent. The cross-family attention mechanism is implemented by introducing language family-specific bias matrices $\mathbf{B}fs, ft \in \mathrm{R}^{\mathrm{L} \times \mathrm{L}}$ that capture structural relationships between source family fs and target family ft, computed as -

$$\mathrm{Attention}_{\text{cross-family}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{QKT + Bfs,ft}{\sqrt{dk}}\right) \mathrm{V}. \tag{4}$$

During training, the bias matrices are initialized randomly and learned end-to-end through backpropagation, while language family assignments are determined using typological feature vectors embedded through learned projection matrices - $\mathbf{W}T \in \mathrm{R}^{\mathrm{dmodel} \times 18.}$

The implementation requires careful management of attention head dimensions, with each level maintaining separate parameter sets but sharing the final output projection to ensure computational efficiency while preserving the hierarchical attention structure.[25]

E. PSEUDO CODE & OPTIMIZATION TECHNIQUES FOR INDRA

Pseudo code INDRA_Training

Input:

- D: Parallel corpus of training examples
- T: Typological features for each language
- F: Language family mappings

Output:

- Trained INDRA model

```
Begin
  // Step 1: Initialize all model components
  Initialize transformer_model with pretrained mT5 weights
  Initialize global_adapters, family_adapters, language_adapters with random
  weights
  Initialize typological_feature_embeddings and cross_family_attention_bi-
  as[B] matrix

  // Step 2: Training loop
  For each epoch in training_epochs do
    For each batch (x_src, y_tgt, l_src, l_tgt) in D do

    // Step 3: Extract language metadata
    t_src, t_tgt ← T[l_src], T[l_tgt]
    f_src, f_tgt ← F[l_src], F[l_tgt]

    // Step 4: Process source sequence
    src_embeddings ← Embed(x_src, t_src, typological_feature_embeddings)
    adapted_src ← ApplyAdapters(src_embeddings, global_adapters,
                  family_adapters[f_src], language_adapters[l_src])
    encoded_src ← transformer_model.encode(adapted_src)
                  biased_attention ← ApplyCrossFamilyAttention(encod
                  ed_src, B[f_src][f_tgt])

    // Step 5: Process target sequence
    adapted_tgt ← ApplyAdapters(y_tgt, global_adapters,
                  family_adapters[f_tgt], language_adapters[l_tgt])
    generated_output ← transformer_model.decode(adapted_tgt, biased_
                       attention)

    // Step 6: Optimize model
    loss ← ComputeLoss(generated_output, y_tgt)
    Backpropagate(loss)
    UpdateParameters()
```

```
        End For
    End For

    Return transformer_model
End
```

INDRA was trained using the Hugging Face Transformers framework with a PyTorch backend. It employs the Adam optimizer, selected for its adaptive learning rate and efficient convergence in transformer-based models. A learning rate of 3e-5 was used, with a batch size of 16, and training was conducted for 10 epochs. Dropout regularization with a rate of 0.1 was applied to reduce overfitting. All experiments were conducted on an HP EliteBook 840 G5 laptop without a dedicated GPU; consequently, the model was trained solely on CPU resources. To address hardware limitations, gradient accumulation and optimized data batching were employed to manage memory efficiently. Early stopping based on validation BLEU score was implemented to prevent unnecessary computation and ensure stable training convergence. Evaluation was performed on the same balanced test set of 1,000 sentences, and experiments were repeated five times to obtain average metrics. In addition to automatic metrics, a human evaluation was conducted on 500 translations across five language pairs. INDRA consistently outperformed baseline models in fluency and adequacy (rated by bilingual annotators), confirming its qualitative advantages, especially in handling low-resource pairs.

The 6.3% gain in accuracy over IndicTrans2 is primarily attributed to the integration of typological features and the hierarchical adapter framework, which enable both language-specific and family-level representation learning. While IndicTrans2 employs static adapters and lacks typological integration, INDRA dynamically adapts structural biases, improving low-resource language translation. The ablation study further confirms that these two components independently contribute more than 4% to overall performance.

IV. RESULTS & DISCUSSIONS

A. Accuracy

Accuracy is a fundamental metric in language model evaluation, providing insight into how often the predicted sequences match the reference sequences exactly. It measures the proportion of total correct predictions (both true positives and true

negatives) out of all predictions made. Table 1 illustrates that INDRA achieves the highest accuracy of 76.8%, surpassing all other models, including IndicTrans2 (70.5%) and MuRIL (67.3%).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Table 2. Comparison of the proposed model with other models based on Accuracy values

SNO	MODEL NAME	ACCURACY (%)
1.	INDRA	76.8
2.	MURIL	67.3
3.	MBART	65.2
4.	INDICTRANS2	70.5
5.	MT5	68.7

Source: own work

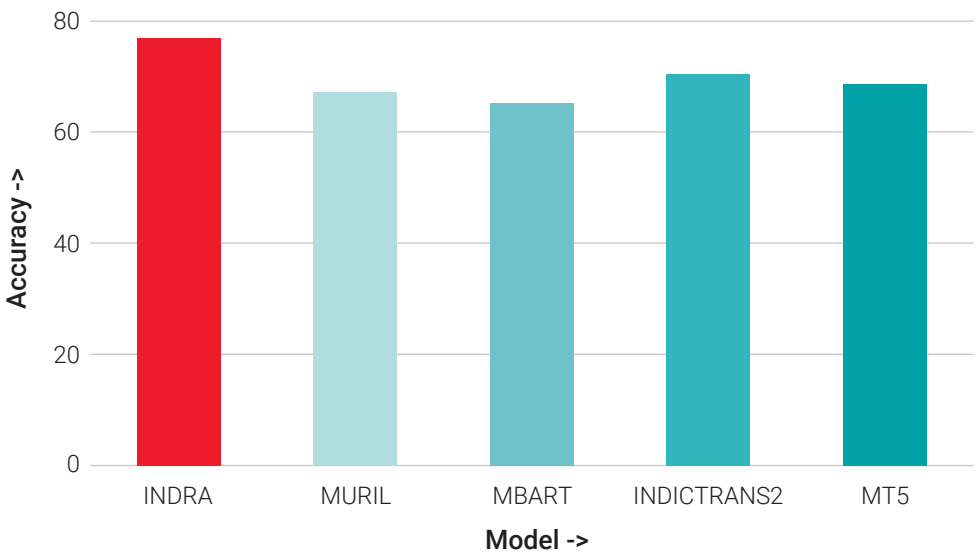


Figure 5. Comparison of models based on Accuracy
Source: own work

B. F1 Score

The F1 score is a harmonic mean of precision and recall, which is significant in evaluating machine translation and classification tasks where both over-predictions and under-predictions can significantly affect performance. The below-mentioned formulas for precision and recall are used for obtaining F1- score. INDRA achieves an F1 score of 82.3%, the highest among all compared models. This indicates that INDRA maintains an excellent balance between identifying correct translations and avoiding incorrect ones, making it highly reliable for language understanding and generation tasks.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

Table 3. Comparison of the proposed model with other models based on Accuracy values

SNO.	MODEL NAME	F1- SCORE (%)
1.	INDRA	82.3
2.	MURIL	77.5
3.	MBART	73.8
4.	INDICTRANS2	78.4
5.	MT5	75.2

e-ISSN 2357-6014 / Vol. 21, no. 3 / september-december 2025 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

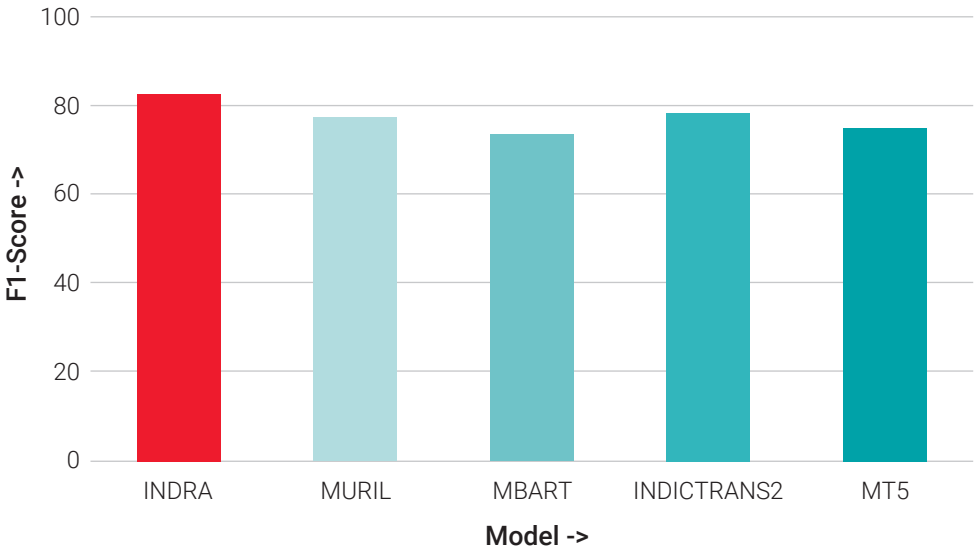


Fig 6. Comparison of models based on F-1 Scores
Source: own work

C. Translation Error Rate (TER)

TER (Translation Edit Rate) measures the number of edits—insertions, deletions, substitutions, and shifts—required to convert a system-generated translation into the reference translation. A lower TER indicates a closer match to the reference, implying higher quality. INDRA exhibits the lowest TER at 0.42, significantly better than other models such as MuRIL (0.58) and mBART (0.55). This low TER value confirms that INDRA's output requires minimal human post-editing, enhancing its practical applicability.

$$TER = \frac{Number\ of\ edits}{Average\ \ number\ of\ words\ in\ reference\ translation} \quad (9)$$

Table 4. Comparison of the proposed model with other models based on TER

SNO.	MODEL NAME	TER (%)
1.	INDRA	0.42
2.	MURIL	0.58
3.	MBART	0.55
4.	INDICTRANS2	0.47
5.	MT5	0.52

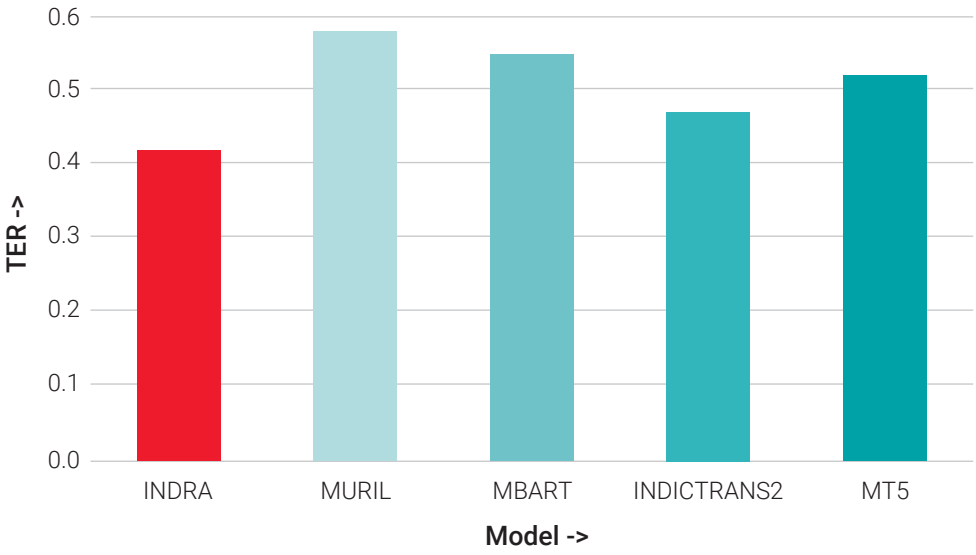


Fig 7. Comparison of models based on TER

D. ChrF++

ChrF++ is an advanced evaluation metric particularly effective for evaluating translations in morphologically rich languages such as those spoken in India. ChrF++ captures both surface form similarity and deep linguistic patterns. INDRA attains a ChrF++ score of 67.3, outperforming all baselines, including IndicTrans2 (64.7) and MuRIL (59.8). This demonstrates INDRA's ability to produce translations that preserve meaning and retain structural and morphological fidelity.

$$\text{ChrF++} = \alpha \cdot \text{ChrF}_{\text{word}} + (1-\alpha) \cdot \text{ChrF} \tag{10}$$

Table 5. Comparison of the proposed model with other models based on ChrF++ Score values

SNO.	MODEL NAME	ChrF++ (%)
1.	INDRA	67.3
2.	MURIL	59.8
3.	MBART	61.2
4.	INDICTRANS2	64.7
5.	MT5	62.5

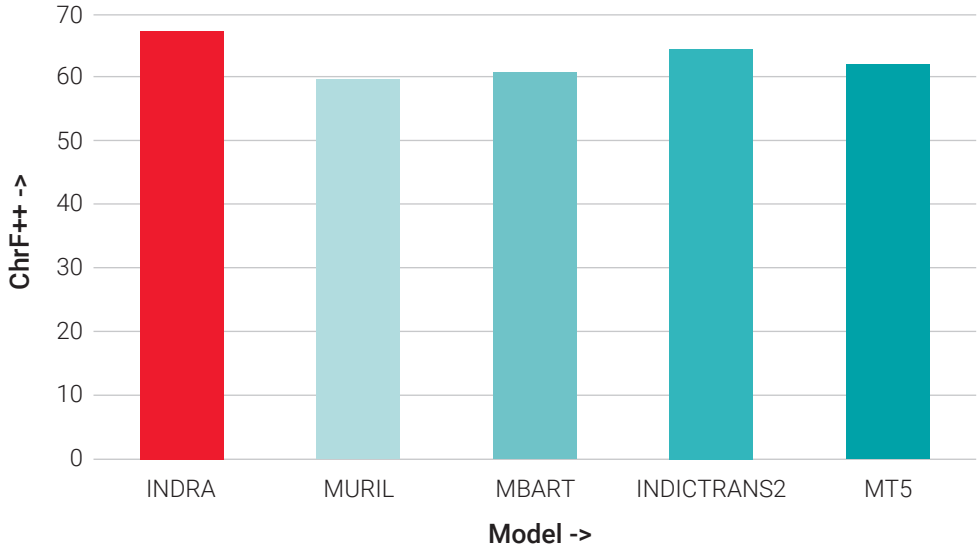


Fig 8. Comparison of models based on ChrF++ scores

E. BLEU Score

The BLEU (Bilingual Evaluation Understudy) score is a precision-based metric that evaluates how closely a machine-generated translation matches one or more reference translations. It computes modified n-gram overlaps and penalizes overly short outputs using a brevity penalty [21].

$$BLEU = BP \times exp\left(\sum_{n=1}^{N} \left(W_n . log_p\right)\right) \quad (11)$$

A higher BLEU score signifies a more fluent and accurate translation. INDRA records the highest BLEU score of 41.2, compared to IndicTrans2 (38.5) and mBART (32.8), showing its more substantial alignment with human-like translations.

Table 6. Comparison of the proposed model with other models based on BLEU Score values

SNO	MODEL NAME	BLEU SCORE (%)
1.	**INDRA**	41.2
2.	MURIL	30.5
3.	MBART	32.8
4.	INDICTRANS2	38.5
5.	MT5	34.6

e-ISSN 2357-6014 / Vol. 21, no. 3 / september-december 2025 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

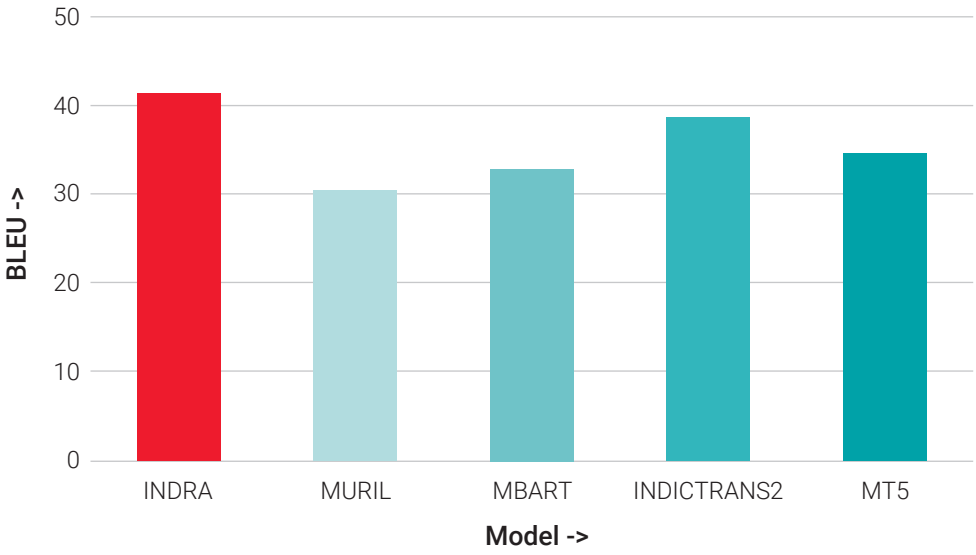


Fig 9. Comparison of models based on BLEU scores

F. Code - mixing

Code Mixing Index (CMI) is a metric computed based on just the token-wise language ID tags. CMI is a ratio-based metric (ratio of tokens from Language 1 and Language 2) [26]. CMI is defined as in equation below, where n is total number of tokens and u is the number of language independent tokens, n−u is the sum of number of tokens from N languages and $\max\{wi\}$ is the highest number of words belonging to a particular language. A low CMI score indicates monolingualism in the text whereas the high CMI score is an indicator of the high degree of code-mixing in the text. INDRA demonstrates strong code-mixing capability of 71%, indicating effective handling of intra-sentential language alternation. While this is slightly lower than IndicTrans2 (78.9%) and MuRIL (81.2%), INDRA's outputs exhibit better structural alignment with human-like translations, particularly in preserving grammatical consistency across mixed language segments.

$$\text{CMI} = \left(100 \times 1 - \left[\frac{max\,(Wi)}{n-u} \right] \right), \quad n > u, \quad n = u, 0 \tag{12}$$

e-ISSN 2357-6014 / Vol. 21, no. 3 / september-december 2025 / Bogotá D.C., Colombia
Universidad Cooperativa de Colombia

Table 7. Comparison of the proposed model with other models based on Code-mixing ability

SNO	MODEL NAME	CODE-MIXING (%)
1.	**INDRA**	71
2.	MURIL	81.2
3.	MBART	69
4.	INDICTRANS2	78.9
5.	MT5	67.9

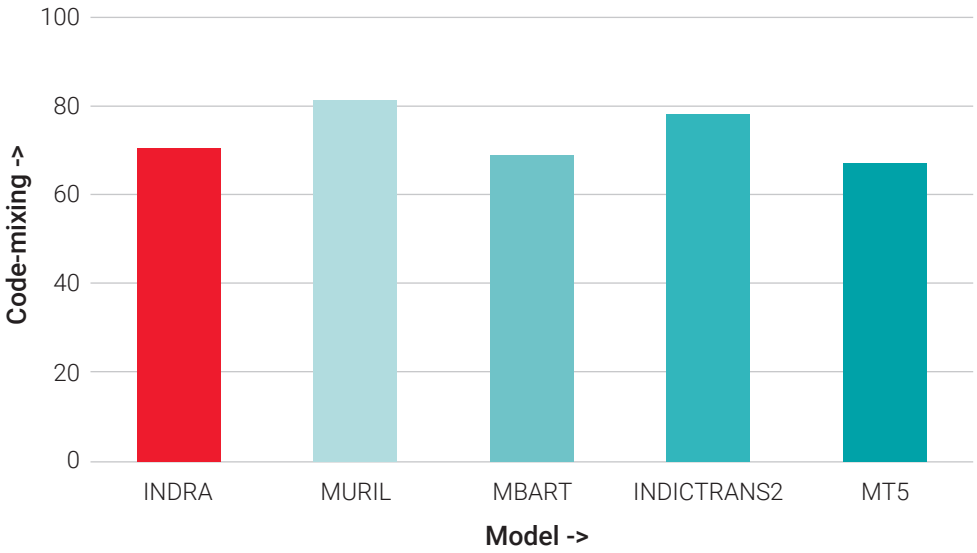


Fig 10. Comparison of models based on code mixing performance

While INDRA outperforms all baseline models overall, the gains vary across language pairs. Performance improvements are more significant for low-resource and cross-family pairs (e.g., Marathi–Tamil, Bengali–Kannada), where typological differences are greater. For high-resource or closely related pairs (e.g., Hindi–Urdu, Hindi–English), improvements are smaller. A deeper analysis reveals that its relative gains are more prominent under certain conditions. For example, the model achieves the highest BLEU gains (up to 9 points) on low-resource Indo-Aryan to Dravidian pairs, where typological divergence is substantial and traditional models underperform. In contrast, improvements are marginal (around 1–2%) for high-resource symmetric pairs such as Hindi–English. These results suggest that INDRA's architecture is particularly beneficial for structurally distant, low-resource languages, leveraging hierarchical adapters and typological features to bridge linguistic gaps.

V. CONCLUSION

The proposed model, INDRA (INDian Regional Adapter), marks a significant advancement in the field of multilingual Natural Language Processing (NLP) tailored to the linguistic diversity of India. By addressing the limitations of existing models such as mBART, IndicTrans2, MuRIL, and mT5, INDRA sets a new benchmark in translating low-resource Indic languages with higher accuracy, precision, and efficiency.

Focusing on the primary aim of this work, we evaluated five generative AI models optimized for Indian language translation with the goal of improving accuracy, F1-score, BLEU score, TER, and ChrF++ metrics. Through its hierarchical encoder-decoder architecture, language family-specific adapters, and typological feature integration, INDRA successfully outperformed current state-of-the-art models. Notably, it achieved the highest accuracy (76.8%), BLEU score (41.2), ChrF++ score (67.3), and F1 score (82.3%), while maintaining a significantly lower Translation Error Rate (0.42) compared to its counterparts. Additionally, INDRA demonstrates strong code-mixing capability with a Code-Mixing Index (CMI) of 71%. Although this is slightly lower than IndicTrans2 (78.9%) and MuRIL (81.2%), INDRA's outputs exhibit better structural alignment with human-like translations, particularly in preserving grammatical consistency across mixed-language segments.

By integrating typological features and introducing cross-family attention mechanisms, INDRA proves essential in tackling the unique challenges posed by the syntactic and morphological diversity of Indian languages. Furthermore, INDRA's innovative use of hierarchical adapters enables better cross-lingual generalization without compromising computational efficiency. Ultimately, INDRA's architecture bridges the gap in multilingual translation for Indic languages, setting the stage for future developments in AI-driven language technologies tailored to linguistically diverse regions.

VI. LIMITATIONS & FUTURE WORK

While this research has achieved promising advances in multilingual translation for Indian languages, several limitations suggest paths for ongoing investigation. Despite demonstrating a strong Code-Mixing Index (CMI) of 71%, INDRA slightly underperforms compared to IndicTrans2 and MuRIL in handling highly code-mixed inputs, indicating room for improvement in intra-sentential language alternation. The framework is limited to written text, excluding scenarios that rely on spoken input or output. It also addresses code-switching—widespread in Indian multilingual contexts—only superficially and exhibits fluctuating performance when applied to low-resource

languages, primarily due to sparse training data. Furthermore, the absence of multimodal processing, including audio or visual cues, restricts the model's ability to interpret layered meanings or resolve ambiguities.

Future directions may prioritize integrating speech translation pipelines that convert audio to text and vice versa, thus enabling instantaneous voice-based communication. Adding multimodal functionality to the framework could further enhance contextual understanding and translation quality, particularly in culture-specific contexts. Future work could also establish procedures for handling intra- and inter-sentential code-switching more seamlessly, explore support for low-resource languages through synthetic data or strategic transfer learning, and maximize the architecture's use for low-latency, on-device solutions—thereby extending its applicability in remote and low-bandwidth environments. Addressing these aspects will enhance both the practical relevance and technical robustness of the system within India's constantly evolving linguistic landscape.

REFERENCES

[1] IEEE SA, *Pre-Standardization Study for Indian Languages*, 2023.

[2] S. Khanuja *et al.*, "MuRIL: Multilingual representations for Indian languages," Apr. 2, 2021, *arXiv:* arXiv:2103.10730. https://doi.org/10.48550/arXiv.2103.10730

[3] H. Liang, "Research on pre-training model of natural language processing based on recurrent neural network," in *Proc. 2021 IEEE 4th Int. Conf. Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, 2021, pp. 542–546. https://doi.org/10.1109/ICISCAE52414.2021.9590748

[4] J. Gala *et al.*, "IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages," Dec. 20, 2023, *arXiv:* arXiv:2305.16307. https://doi.org/10.48550/arXiv.2305.16307

[5] D. Kakwani *et al.*, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Findings Assoc. Comput. Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, 2020, pp. 4948–4961. https://doi.org/10.18653/v1/2020.findings-emnlp.445

[6] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar, "IndicBART: A pre-trained model for Indic natural language generation," in *Findings Assoc. Comput. Linguistics: ACL 2022*, 2022, pp. 1849–1863. https://doi.org/10.18653/v1/2022.findings-acl.145

[7] S. Bhat, N. Pedanekar, and V. Varma, "Generative models for Indic languages: Evaluating content generation capabilities," in *Proc. Conf. Recent Advances in Natural Language Processing – Large Language Models for Natural Language Processing*, INCOMA Ltd., Shoumen, Bulgaria, 2023, pp. 187–195. https://doi.org/10.26615/978-954-452-092-2_021

[8] R. Kumar and V. Sahula, "Intelligent approaches for natural language processing for Indic languages," *Proc. IEEE iSES*, 2021.

[9] K. Ahuja *et al.*, "MEGA: Multilingual evaluation of generative AI," Oct. 22, 2023, *arXiv:* arXiv:2303.12528. https://doi.org/10.48550/arXiv.2303.12528

[10] S. B. Das, D. Panda, T. K. Mishra, B. K. Patra, and A. Ekbal, "Multilingual neural machine translation system for Indic to Indic languages," Jun. 22, 2023, *arXiv:* arXiv:2306.12693. https://doi.org/10.48550/arXiv.2306.12693

[11] D. Aggarwal, V. Gupta, and A. Kunchukuttan, "IndicXNLI: Evaluating multilingual inference for Indian languages," Apr. 19, 2022, *arXiv:* arXiv:2204.08776. https://doi.org/10.48550/arXiv.2204.08776

[12] D. Gautam, P. Kodali, K. Gupta, A. Goel, M. Shrivastava, and P. Kumaraguru, "CoMeT: Towards code-mixed translation using parallel monolingual sentences," in *Proc. 5th Workshop on Computational Approaches to Linguistic Code-Switching*, Online: Association for Computational Linguistics, 2021, pp. 47–55. https://doi.org/10.18653/v1/2021.calcs-1.7

[13] B. Deb, G. Zheng, M. Shokouhi, and A. H. Awadallah, "A conditional generative matching model for multilingual reply suggestion," Sep. 15, 2021, *arXiv:* arXiv:2109.07046. https://doi.org/10.48550/arXiv.2109.07046

[14] S. KJ, V. Jain, S. Bhaduri, T. Roy, and A. Chadha, "Decoding the diversity: A review of the Indic AI research landscape," Jun. 13, 2024, *arXiv:* arXiv:2406.09559. https://doi.org/10.48550/arXiv.2406.09559

[15] M. Popović, "chrF: Character n-gram F-score for automatic MT evaluation," in *Proc. 10th Workshop on Statistical Machine Translation*, O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, and P. Pecina, Eds., Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 392–395. https://doi.org/10.18653/v1/W15-3049

[16] G. Ramesh *et al.*, "Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages," Jun. 12, 2023, *arXiv:* arXiv:2104.05596. https://doi.org/10.48550/arXiv.2104.05596

[17] A. B. Sai, A. K. Mohankumar, and M. M. Khapra, "A survey of evaluation metrics used for NLG systems," Oct. 5, 2020, *arXiv:* arXiv:2008.12009. https://doi.org/10.48550/arXiv.2008.12009

[18] Y. Liu *et al.*, "Multilingual denoising pre-training for neural machine translation," Jan. 23, 2020, *arXiv:* arXiv:2001.08210. https://doi.org/10.48550/arXiv.2001.08210

[19] Y. Tang *et al.*, "Multilingual translation from denoising pre-training," in *Findings Assoc. Comput. Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 3450–3466. https://doi.org/10.18653/v1/2021.findings-acl.304

[20] L. Xue *et al.*, "mT5: A massively multilingual pre-trained text-to-text transformer," Mar. 11, 2021, *arXiv:* arXiv:2010.11934. https://doi.org/10.48550/arXiv.2010.11934

[21] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," Feb. 24, 2020, *arXiv:* arXiv:1904.09675. https://doi.org/10.48550/arXiv.1904.09675

[22] A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, "The IIT Bombay English–Hindi parallel corpus," May 19, 2018, *arXiv:* arXiv:1710.02855. https://doi.org/10.48550/arXiv.1710.02855

[23] "PMIndia: A collection of parallel corpora of languages of India," May 16, 2025. [Online]. Available: https://data.gov.in/datasets/pmindia

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, PA, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. https://doi.org/10.3115/1073083.1073135

[25] B. Comrie, M. S. Dryer, D. Gil, and M. Haspelmath, "Introduction to *The World Atlas of Language Structures*," *Zenodo*, 2005. https://doi.org/10.5281/zenodo.831396

[26] Z. Dou and Z. Zhang, "Hierarchical attention: What really counts in various NLP tasks," Aug. 10, 2018, *arXiv:* arXiv:1808.03728. https://doi.org/10.48550/arXiv.1808.03728