****

**LINGO SENSE:AI for Translating Colloquial Languages**

****

**Student Name&ID**

**Keerthana Bollepally(22WU01040287)**

**Alaka Asritha Ghantasala(22WU0104129)**

**Kamal Sai Cherukuri(22WU0104030)**



**Prof. Dr. Uday Chandra**

****

**Department of Computer Science and Engineering**

**School of Technology Woxsen University**

**Hyderabad**

**2025-2026**

**Table of Contents**

1. **Abstract**
2. **Introduction**  
   2.1 Problem  
   2.2 Background Information / Statement  
   2.3 Significance and Scope
3. **Literature Review**  
   3.1 Literature Review Summary  
   3.2 Limitations or Research Gap  
     3.2.1 Clear Articulation  
     3.2.2 Justification  
     3.2.3 Evidence-Based Support  
     3.2.4 Categorization of Gaps
4. **Technology Review**  
   4.1 Technology Review Table  
   4.2 Technology Review Summary  
     4.2.1 Key Findings  
     4.2.2 Gaps and Limitations  
     4.2.3 Comparative Analysis  
     4.2.4 Justification for LingoSense
5. **Objectives**  
   5.1 SMART Criteria  
   5.2 Project-Specific Notes
6. **Methodology**  
   6.1 Approach and Implementation Process  
   6.2 Tools and Technologies  
   6.3 Block Diagram and System Architecture  
     6.3.1 Key Components  
     6.3.2 Component Interactions  
     6.3.3 Data Flow Highlights  
   6.4 Project Timeline and Milestones
7. **Novelty**  
   7.1 Unique Contribution  
   7.2 Gap Addressing  
   7.3 Technical Advancement  
   7.4 Industry / Societal Impact  
   7.5 Evidence of Novelty
8. **Results / Outcomes**  
   8.1 Achieved Outcomes  
   8.2 Data and Evidence  
   8.3 Technical Achievements  
   8.4 Practical Implications
9. **Conclusion**  
   9.1 Summary of Key Findings  
   9.2 Significance and Impact  
   9.3 Limitations  
   9.4 Lessons Learned
10. **Recommendations and Future Scope**  
    10.1 Improvements for Capstone Project-II  
    10.2 Research and Industrial Applications  
    10.3 Long-Term Vision
11. **Acknowledgment**
12. **References**

**1.Abstract**

LingoSense: AI for Translating Colloquial Languages focuses on enhancing translation accuracy for informal, code-mixed, and colloquial text frequently used in multilingual Indian communication. The project addresses the limitations of conventional translators like Google Translate, which often fail to interpret transliterated and slang-rich content, leading to context loss.

The system primarily concentrates on translation and code-mix handling, employing transformer-based models such as FLAN-T5, NLLB-200, IndicTrans2, and RLM-Hinglish Translator, integrated through Hugging Face Transformers and IndicTransToolkit. The implementation follows a structured pipeline involving transliteration, code-mix detection, text normalization, and multilingual translation across six major Indian languages—Hindi, Telugu, Tamil, Bengali, Marathi, and Malayalam. A Streamlit interface enables real-time, user-interactive translation.

Experimental evaluations indicate improved contextual understanding and translation fluency compared to traditional systems. By effectively preserving cultural tone and informal expressions, LingoSense supports inclusivity in digital communication, education, and cross-lingual interaction, making it valuable for industries involving regional content, conversational AI, and social media localization.

**2.Introduction**

Language is a vital medium for expressing culture, emotion, and identity. In India’s multilingual context, communication often involves code-mixing—the blending of English with regional languagesespecially across social media and messaging platforms, where users use transliteration and informal expressions.

Conventional translation systems like Google Translate and Microsoft Translator struggle with such inputs, leading to inaccurate or culturally irrelevant translations.

To overcome this, LingoSense was developed as an AI-driven multilingual translation framework designed for code-mixed and colloquial text. It integrates transformer-based models such as FLAN-T5, NLLB-200, IndicTrans2via Hugging Face and IndicTransToolkit. The core pipeline performs transliteration, code-mix detection, normalization, and multilingual translation across six Indian languages—Hindi, Telugu, Tamil, Bengali, Marathi, and Malayalam.

Implemented in Python (Jupyter Notebooks) and deployed using Streamlit, LingoSense emphasizes context preservation to retain emotional and cultural tones. Experimental evaluation showed strong fluency, coherence, and adaptability to informal linguistic inputs.

By bridging the gap between formal translation systems and real-world language use, LingoSense fosters linguistic inclusivity and regional language empowerment, with applications in education, digital communication, and AI-driven customer interaction.

**2.1 Problem Statement**

Despite major advances in NLP and machine translation, existing systems such as Google Translate, Microsoft Translator, and DeepL still struggle with code-mixed and colloquial Indian languages. Trained mainly on formal datasets, these models fail to capture the informal, blended nature of everyday communication. Users often mix English with regional languages e.g., *“nenu class ki vellali”* (Telugu-English) or *“main school ja rahi hoon”* (Hindi-English) and use inconsistent transliterations like *“namaste”* and *“namasthe”*, causing translation errors and context loss.

LingoSense addresses these issues by integrating advanced Transformer-based models—FLAN-T5, IndicTrans2, NLLB-200, and RLM-Hinglish Translator—to improve translation fluency, preserve emotion and context, and handle informal expressions and slang. Ultimately, it aims to enable seamless multilingual communication across India’s diverse linguistic landscape, supporting applications in social media, education, customer support, and conversational AI.

**2.2 Background Information**Language translation is a crucial NLP field bridging linguistic and cultural gaps. In multilingual India, effective translation is vital for education, governance, and digital communication. Most systems, however, perform well only on formal text and struggle with **transliterated and code-mixed language** common in daily use.

1. Relevant Domain Knowledge  
Machine Translation evolved from rule-based and statistical models to Neural Machine Translation (NMT) with encoder–decoder and attention mechanisms. Transformer models like BERT, T5, mBART, MarianMT, and NLLB improve fluency and accuracy. Indian languages require context-aware systems to handle code-mixing, transliteration, and informal syntax while preserving meaning and culture.

2. Current Industry Trends  
Popular translators like Google Translate and DeepL rely on structured data, performing poorly on informal or bilingual text. Indian initiatives such as AI4Bharat, IndicNLP, and Bhashini advance Indic translation but still face challenges with transliteration and sentiment retention, highlighting the need for culturally adaptive, transliteration-integrated NLP frameworks.

3. Historical Context  
Translation systems evolved through:  
• Rule-Based (1950s–1990s): Grammar-focused but rigid.  
• Statistical (1990s–2010s): Used bilingual corpora but lacked contextual depth.  
• Neural (2015–Present): Transformer-based systems improved fluency and context.  
Despite progress, code-mixed and low-resource Indian languages remain underrepresented due to limited datasets and transliteration inconsistencies.

4. Technological Evolution  
Transformers revolutionized multilingual NLP with models like MarianMT, M2M-100, and NLLB-200, while IndicTrans2 and Indic NLP tools enhanced Indian language processing. Recent methods like Retrieval-Augmented Generation (RAG) and LoRA fine-tuning enable better adaptation to low-resource data. LingoSense builds on these advancements, combining multiple Transformer models for improved transliteration, code-mix handling, and cultural context preservation.

**2.3 Significance and Scope**

LingoSense plays a crucial role in advancing multilingual AI by enabling accurate processing of code-mixed, transliterated, and regional Indian text. Using IndicTrans2, NLLB-200, FLAN-T5, and RLM-Hinglish Translator, it provides high-quality transliteration and translation for Hindi-English (Hinglish) inputs.

The system benefits educators, researchers, language learners, and digital platforms like Twitter and WhatsApp, while supporting public services and inclusivity in India’s multilingual ecosystem. Socially, it promotes linguistic equality; educationally, it aids bilingual learning; and technologically, it strengthens Indic NLP research and dataset quality.

A Streamlit-based interface offers real-time translation and visualization. Evaluated using BLEU and accuracy metrics, LingoSense shows strong transliteration and translation performance, with future expansion planned for Tamil-English and Bengali-English. Through its structured development process data collection, training, evaluation, and deployment LingoSense enhances multilingual accessibility and supports India’s growing digital communication landscape.

**3.Literature Review**



**3.1 Literature Review Summary**

The literature highlights the evolution and challenges of Neural Machine Translation (NMT), focusing on low-resource, multilingual, and context-aware systems.

1. Common Research Trends:  
Studies emphasize encoder–decoder architectures with attention, low-resource translation via transfer and unsupervised learning, and multilingual/context-aware NMT. Transformer models like BERT, T5, and mBART improve fluency and scalability.

2. Research Gaps:  
Challenges include scarce resources for morphologically rich languages, lack of context-aware evaluation, limited multimodal AI integration, high computational cost, and insufficient Indian multilingual datasets.

3. Theoretical Foundation:  
NMT relies on sequence-to-sequence and Transformer architectures (Vaswani et al., 2017). Transfer learning and multilingual embeddings support cross-lingual adaptation, while contextual learning ensures semantic consistency over longer text.

4. Methodological Insights:  
Unsupervised/dual learning, back-translation, and multilingual pretraining are common. Transformers outperform RNNs/CNNs, and model compression improves efficiency.

5. Limitations and Challenges:  
Key issues include data scarcity, overfitting, high computational demands, weak contextual evaluation (BLEU, TER), and limited cultural/linguistic representation.

6. Justification for Research:  
The literature highlights the need for context-aware, resource-efficient multilingual NMT. LingoSense addresses this via RAG and multilingual models (mT5, IndicBERT), supporting Bhashini’s inclusive language goals.

**3.2 Limitations or Research Gap**

Despite significant advances in Neural Machine Translation, current models still underperform when handling code-mixed, transliterated, and informal text commonly found in Indian languages

.**3.2.1Clear Articulation:**Models like mBART, NLLB-200, and IndicTrans2 excel on clean data but fail with informal or transliterated inputs (Dabre et al., 2022; Wang et al., 2021). They show weak adaptation in low-resource settings and sentence-level contextual inconsistency (Castilho & Knowles, 2024; Maruf et al., 2021). Back-translation partially mitigates these issues (Gibadullin et al., 2022; Ranathunga et al., 2023), leaving a gap for hybrid, context-aware systems handling transliteration and code-mixing.

**3.2.2 Justification**

LingoSense bridges this gap by integrating phonetic-aware transliteration and context-preserving translation using IndicTrans2, NLLB-200, and Flan-T5 in a single modular pipeline.  
This approach:

* Combines script conversion and semantic alignment.
* Enhances contextual accuracy for informal, code-mixed inputs.
* Offers scalability for multilingual and social media contexts.

**3.2.3 Evidence-Based Support**

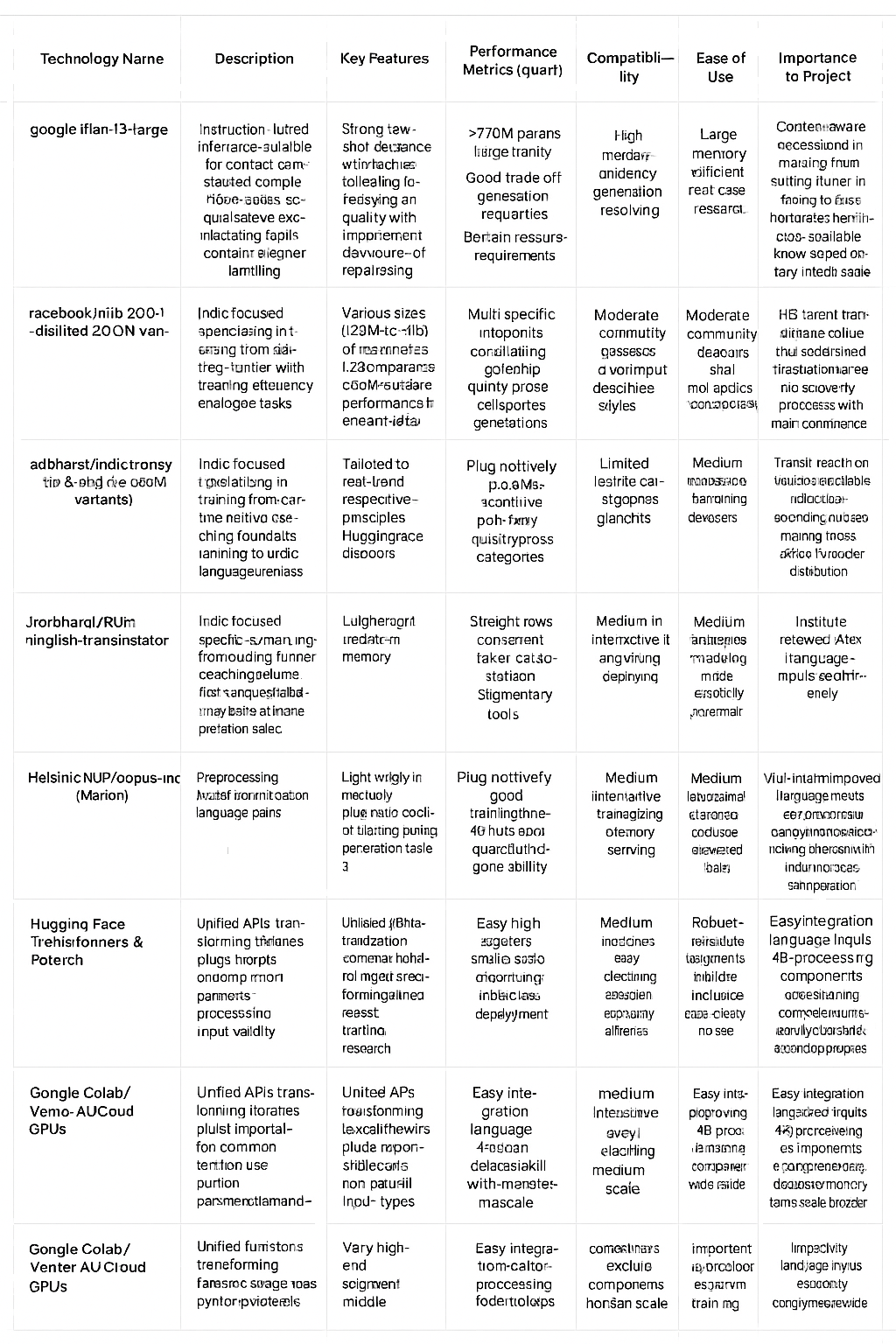
* Wang et al. (2021) and Haddow et al. (2021) confirm limited low-resource data.
* Castilho & Knowles (2024) and Maruf et al. (2021) identify weak context modeling.
* Gibadullin et al. (2022) and Chu (2021) highlight domain adaptation limits.
* Verma & Bhattacharyya (2020) note the absence of transliteration-integrated systems.

These studies validate the need for LingoSense’s hybrid approach.

**3.2.4 Categorization of Gaps**

* Technical: Limited scalability and poor handling of code-mixed inputs (Wang et al., 2021).
* Functional: Lack of transliteration integration (Verma & Bhattacharyya, 2020).
* Methodological: Overreliance on parallel corpora; weak phonetic adaptation (Chu, 2021).
* Usability: Minimal user-centric or interactive designs (Dabre et al., 2022).
* Cost: High computational expense limits accessibility (Stahlberg, 2021).
* Security: Gaps in privacy and bias mitigation (Wang et al., 2021).
* Research: Absence of context-aware, transliteration-integrated NMT for Indian languages (Ranathunga et al., 2023).

**4.Technology Review**

**Table 4.1: Technology Review — LingoSense (selected technologies & tools)**

**4.2 Technology Review Summary**

Existing translation systems like Google Translate and DeepL perform well on formal text but struggle with code-mixed, transliterated, or informal inputs. Trained mainly on structured corpora, they fail to capture regional slang, phonetic spellings, and cultural nuances. Transformer-based models such as mBART, IndicTrans, and NLLB improved multilingual translation but still lack adaptation for mixed-script and informal text. There remains a technological gap in unifying transliteration and translation into a single efficient pipeline.

LingoSense bridges this gap by implementing a custom transliteration-to-translation mechanism that effectively handles mixed-script inputs, preserves context and sentiment, and enhances cross-lingual communication, education, and inclusivity.

**4.2.1 Key Findings**

* Transformer models like mBART, NLLB, and IndicTrans have advanced multilingual translation through attention mechanisms and large-scale pretraining.
* Preprocessing tools such as IndicTransToolkit and Sanscript are essential for transliteration and text normalization in Indic NLP pipelines.
* Specialized models like RLM-Hinglish Translator are crucial for informal and code-mixed data, highlighting the need for context-aware translation.

**4.2.2 Gaps and Limitations**

1. Poor handling of code-mixed and transliterated text due to formal training corpora.
2. Limited contextual and cultural understanding—tone and idioms often lost.
3. No unified transliteration–translation pipeline, leading to higher error rates.
4. Minimal domain adaptation for social media or conversational data.
5. Lack of annotated informal datasets, reducing robustness for code-switched inputs.

These gaps highlight the need for a context-aware, multi-model framework tailored to India’s linguistic diversity.

**4.2.3 Comparative Analysis**

Each translation model offers distinct strengths but also limitations, reinforcing the hybrid approach of LingoSense.

* NLLB-200 ensures wide multilingual coverage but struggles with informal text.
* IndicTrans2 performs well on native-script data but not on Romanized input.
* RLM-Hinglish Translator handles Hinglish effectively yet remains limited in scope.
* FLAN-T5 enhances fluency and readability but is computationally heavy.
* Preprocessing tools like IndicTransToolkit and Sanscript reduce transliteration errors.

Combining NLLB for coverage, IndicTrans2 for precision, RLM-Hinglish for code-mix handling, and FLAN-T5 for fluency enables LingoSense to produce accurate, context-aware, and culturally coherent translations.

**4.2.4 Innovation Opportunities**

LingoSense introduces a unified transliteration-to-translation pipeline that minimizes information loss and processes both Romanized and native-script inputs efficiently. It employs context-aware ensemble modeling and post-processing using FLAN-T5 to improve fluency and idiomatic expression.

Practical strategies include:

* Low-resource optimization: LoRA and PEFT fine-tuning for limited data adaptation.
* Use of informal datasets: Incorporating real-world social media and chat data for better contextual understanding.

These innovations enhance colloquial fluency, contextual accuracy, and transliteration awareness, making LingoSense socially relevant and technically robust.

**4.2.5 Justification for LingoSense**

LingoSense tackles key technological and linguistic challenges through a multi-model ensemble combining NLLB-200, IndicTrans2, RLM-Hinglish, and FLAN-T5. Its preprocessing pipeline with IndicTransToolkit and Sanscript ensures accurate Romanized text normalization before translation. Unlike mainstream formal-text models, LingoSense is optimized for informal, real-world, code-mixed content, capturing the nuances of bilingual digital communication. Post-processing enhances fluency, coherence, and cultural relevance.

By integrating these capabilities, LingoSense stands out as a practical, impactful, and inclusive innovation in Indic NLP, advancing linguistic accessibility and digital inclusivity across India.

**5.Objectives**

The LingoSense project aims to develop a robust AI-driven framework for processing transliterated, code-mixed, and informal Indian language text. It focuses on delivering contextually accurate outputs and improving multilingual accessibility through an integrated transliteration-to-translation pipeline. The objectives are defined using the SMART (Specific, Measurable, Achievable, Relevant, and Time-bound) framework for structured execution and measurable outcomes.

* **To develop a hybrid transliteration–translation framework** using Transformer-based models such as *IndicTrans2, NLLB-200, FLAN-T5,* and *RLM-Hinglish*, prioritizing transliteration accuracy and informal text handling across at least five Indian languages, with a prototype ready in six weeks.
* **To achieve ≥85% transliteration and contextual accuracy**, evaluated using BLEU, COMET, and transliteration fidelity on real-world conversational datasets.
* **To implement a Streamlit-based web platform** for real-time transliteration and context-aware translation of Romanized and mixed-script text, ensuring tone and sentiment preservation by Week 10.
* **To fine-tune and optimize models** on a curated dataset of 20,000+ code-mixed and transliterated samples, collected through **web scraping** from social media, forums, and online text sources, enhancing model adaptability to phonetic variations and colloquial expressions within eight weeks.
* **To benchmark performance** against *Google Translate, DeepL,* and *Microsoft Translator*, documenting evaluation results by Week 14.

**5.1 SMART Criteria**

* **Specific:**  
  Develop an AI-based system for accurate processing of code-mixed and transliterated Indian text using Transformer models (*IndicTrans2, NLLB-200, FLAN-T5,* and *RLM-Hinglish*), focusing on Hindi-English (Hinglish) and regional variants.
* **Measurable:**  
  Evaluate performance using BLEU, COMET, and transliteration accuracy, targeting ≥85% accuracy on real-world datasets. Demonstrate usability through a Streamlit-based interactive platform.
* **Achievable:**  
  Implementation is feasible within the semester using open-source pre-trained models and cloud tools like Google Colab for fine-tuning and deployment.
* **Relevant:**  
  LingoSense addresses the gap in current translation systems unable to process informal, mixed-script inputs, promoting linguistic inclusivity and multilingual accessibility.
* **Time-bound:**  
  A structured 14-week plan ensures steady progress:
  + **Weeks 1–3:** Framework design, data collection, preprocessing
  + **Weeks 4–8:** Model integration and fine-tuning
  + **Weeks 9–11:** Testing and optimization
  + **Weeks 12–14:** Validation, documentation, and Streamlit deployment

**5.2 Project-Specific Notes**

**LingoSense** tackles the limitations of current multilingual translation systems that struggle with code-mixed and informal Indian text. It focuses on Hindi-English (Hinglish) and regional languages through four key objectives:

1. **Multilingual Transformer Integration:**  
   Fine-tunes *IndicTrans2, NLLB-200, RLM-Hinglish,* and *FLAN-T5* within a unified pipeline to ensure accurate, culturally coherent translation across scripts.
2. **Transliteration and Preprocessing:**  
   Uses *IndicTransToolkit* and *Sanscript* for script normalization, improving phonetic accuracy and handling colloquial and mixed-script variations.
3. **Performance Evaluation:**  
   Benchmarked using BLEU, COMET, and accuracy metrics, targeting ≥85% fidelity for code-mixed, informal datasets.
4. **User-Interactive Platform:**  
   A Streamlit interface enables users to input code-mixed text and view transliterations and translations in real time, demonstrating practical usability.

By emphasizing transliteration-aware processing, contextual understanding, and accessibility, **LingoSense** advances Indic NLP research while promoting multilingual inclusivity and effective digital communication across India’s diverse linguistic landscape.

**6. Methodology**The LingoSense project adopts a structured, research-driven implementation methodology aimed at developing a robust multilingual translation framework capable of accurately handling code-mixed and transliterated Indian languages. The methodology combines model-based experimentation, comprehensive data preprocessing, and systematic system design into a cohesive workflow, ensuring both technical rigor and practical feasibility.

The implementation process is organized into distinct, sequential phases, including data preparation, model selection, model integration, fine-tuning, evaluation, and deployment. Each phase is carefully designed to address the challenges posed by Romanized text, code-switching, and informal language patterns, ensuring that the system can reliably process mixed-script inputs and generate contextually accurate translations.

**6.1 Approach and Implementation Process**

The LingoSense project uses a research-driven approach, combining advanced NLP and Transformer-based multilingual models to process code-mixed and transliterated Indian languages. The focus is on accurate transliteration and mixed-script handling, with translation enhancing contextual fluency. The system emphasizes modular design, iterative testing, and validation for efficiency, scalability, and user accessibility.

**1. System Design and Architecture**  
LingoSense follows a modular pipeline with four components:

* Input Processing: Users enter code-mixed or transliterated text (e.g., Romanized Hindi-English). Text is normalized, tokenized, and prepared using IndicTransToolkit and Sanscript for phonetic accuracy.
* Transliteration Module: Converts Roman-script/mixed inputs to native scripts. IndicTrans2 and NLLB-200 handle translation, while RLM-Hinglish Translator processes Romanized code-mixed text.
* Contextual Refinement: FLAN-T5 improves fluency, readability, and tone preservation.
* Output Visualization: Streamlit displays real-time transliterations and translations for comparison.

This modular design ensures flexibility, interoperability, and easy scalability for additional languages.

**2. Development Phases**

1. **Planning & Design:**
   * Reviewed existing multilingual systems and identified limitations in code-mixed handling.
   * Designed a modular architecture emphasizing transliteration accuracy.
2. **Implementation:**
   * Integrated *IndicTrans2, NLLB-200, RLM-Hinglish Translator,* and *FLAN-T5* into a unified pipeline.
   * Built preprocessing scripts using *Indic NLP Library* and *Sanscript* for normalization and transliteration.
3. **Testing & Optimization:**
   * Conducted iterative testing on real-world and web-scraped social media datasets.
   * Fine-tuned models for improved transliteration accuracy and contextual fluency.
4. **Deployment & Documentation:**
   * Developed a *Streamlit* interface for real-time demonstration.
   * Documented system performance and transliteration quality improvements.

**3. Testing and Validation Strategies**

To ensure reliability and transliteration quality:

* **Quantitative Evaluation:** Metrics like *BLEU, COMET,* and transliteration accuracy measured mixed-script performance.
* **Comparative Analysis:** Model outputs compared to determine optimal configurations.
* **User Validation:** Human evaluators reviewed tone, semantic correctness, and script accuracy.
* **Error Analysis:** Logged and resolved issues like mismatched transliterations and script-switching errors.

**4. Quality Assurance Procedures**

Maintained high standards through:

* **Version Control:** Managed via *GitHub* for traceability.
* **Automated Testing:** Python pipelines verified transliteration and mixed-script accuracy.
* **Performance Monitoring:** Tracked accuracy (>85%) and contextual fidelity.
* **Feedback Integration:** User feedback refined model outputs and UI presentation.

**5. Data Collection and Analysis Methods**

Robust data handling ensured adaptability to informal and code-mixed text:

* **Sources:** Data collected from *AI4Bharat, IndicCorp, Hinglish Wikipedia,* and web-scraped social media content.
* **Cleaning & Annotation:** Removed noise and inconsistencies using Python regex scripts.
* **Augmentation:** Generated synthetic code-mixed samples to improve generalization.
* **Analysis:** Performed token frequency and transliteration pattern analysis to reduce dataset bias.

**6.2 Tools and Technologies**

The **LingoSense** system used a combination of programming languages, frameworks, and development tools for efficient, scalable, and reproducible implementation.

1. **Programming Languages:**
   * **Python:** Core language for model integration, preprocessing, and pipeline development.
   * **JavaScript (optional):** Used in *Streamlit* for interactive visualization.
2. **Frameworks and Libraries:**
   * *Hugging Face Transformers* – For multilingual model integration and fine-tuning.
   * *PyTorch* – Deep learning framework for training and inference.
   * *Indic NLP Library* & *Sanscript* – For transliteration and text normalization.
   * *Streamlit* – For real-time interface development.
   * *NumPy, Pandas, Scikit-learn* – For preprocessing and analysis.
3. **Hardware Components:**
   * Cloud-based GPU environments (*Google Colab*) used for fine-tuning; no physical hardware required.
4. **Software Tools:**
   * *Google Colab* (training), *GitHub* (version control), *Jupyter/VS Code* (development), and *SQLite/JSON* (lightweight storage).
5. **Statistical and Analytical Tools:**
   * *BLEU, COMET,* and accuracy metrics for quantitative evaluation.
   * Custom Python scripts for dataset validation and transliteration checks.
6. **Development Methodology:**
   * **Agile Approach:** Iterative cycles for preprocessing, integration, testing, and refinement, ensuring consistent improvements in code-mixed handling.

**6.3 Block Diagram and System Architecture**

* The LingoSense architecture is designed as a modular, end-to-end pipeline to efficiently process code-mixed, transliterated, and informal Indian language inputs. It integrates multiple Transformer-based models and preprocessing tools to perform transliteration, translation, and contextual refinement seamlessly from input to output.

**6.3.1 Key Components**

* **User Input Interface:**

Accepts Romanized Hindi, Hinglish, or other Indian language text containing informal expressions, abbreviations, or mixed scripts.

* **Input Processing Module:**

Normalizes and tokenizes text before transliteration.

Uses *IndicTransToolkit* and *Sanscript* to convert Romanized text to native scripts, ensuring consistency for downstream processing.

* **Transliteration Module:**

Converts Roman-script text into native Indic scripts with phonetic accuracy while preserving informal tone and style.

* **Translation Module:**

Processes transliterated text using multilingual models like *IndicTrans2* and *NLLB-200* for Indic–Indic and Indic–English translation.

RLM-Hinglish Translator handles direct Romanized, code-mixed inputs.

* **Contextual Refinement Module:**

Employs *Flan-T5* for post-processing to enhance fluency, grammar, and contextual accuracy, maintaining tone and naturalness.

* **Output Visualization Module:**

Displays transliteration and translation results via a *Streamlit* interface, allowing real-time comparison of inputs and outputs.

* **Data Flow:**

Input → Preprocessing → Transliteration → Translation → Refinement → Output Display

Each stage manages informal expressions and script variations, minimizing errors across the pipeline**.**

**6.3.2 Component Interactions**

Modules interact sequentially in a linear, modular workflow—each passing processed data to the next.

Users engage only with the Input and Output interfaces; all intermediate processing occurs automatically within the backend.

**6.3.3 Data Flow Highlights**

* Supports real-time transliteration of Romanized text.
* Handles code-mixed sentences while preserving informal tone and meaning.
* Dynamically selects translation models based on input type (native or Romanized).

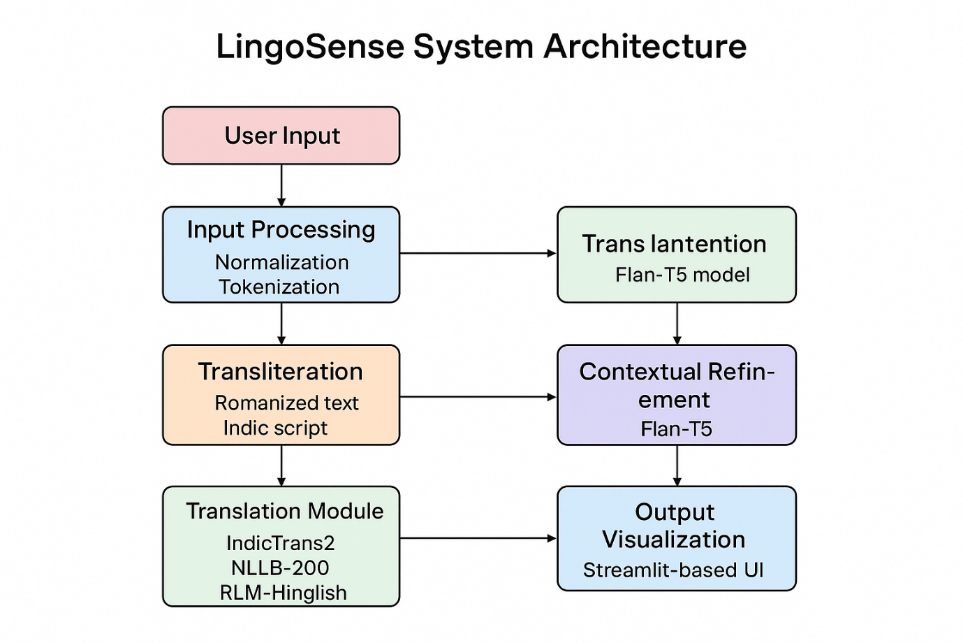


Figure 1: System Architecture

**6.4.Project Timeline and Milestones**

The development of LingoSense is planned over a 12-week timeline, organized into four main phases. Each phase includes specific milestones, deliverables, dependencies, and resource allocation. The timeline ensures iterative progress while allowing for testing, optimization, and deployment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Start Date | End Date | Milestones | Dependencies | Resource Allocation |
| **1. Planning & Design** | Week 1 | Week2 | Review of multilingual frameworks,  System design,  Dataset preparation | None | 2 NLP,1 project lead |
| **2. Data Preparation** | Week 3 | Week 4 | Data collection from multiple sources,  Cleaning, annotation, and augmentation,  Preprocessing script development | Completion of system design | 2 Data Engineers, 1 NLP researcher |
| **3. Model Selection & Integration** | Week4 | Week 9 | Model selection: IndicTrans2, NLLB-200, RLM-Hinglish, Flan-T5.  Pipeline integration  Preprocessing & normalization modules | Completion of data preparation | 2 NLP Engineers, 1 QA Engineer |
| **4. Model Fine-tuning & Testing** | Week10 | Week 11 | Testing on real-world datasets.  Model fine-tuning.  Error analysis & optimization and website creation | Completion of model integration | 2 NLP Engineers, 1 QA Engineer |
| **5. Documentation & Final Review** | Week 12 | Week 13 | Report preparation .Performance evaluation.  Project submission  on of project deliverables | Completion of deployment | 1 Project Lead, 1 NLP researcher |

**7. Novelty**

The LingoSense project introduces a novel framework for processing code-mixed and transliterated Indian languages, addressing limitations in current multilingual translation systems that struggle with informal, mixed-script inputs. Its innovation lies in integrating transliteration, translation, and contextual refinement into a unified pipeline that ensures phonetic accuracy, contextual relevance, and tone preservation. Designed for scalability, it supports multiple languages, transliteration schemes, and real-time interaction through a Streamlit interface. Data-driven preprocessing and augmentation further enhance robustness against informal spellings, abbreviations, and script variations.

Key Innovative Features:

* Hybrid Transliteration–Translation Pipeline: Converts Romanized or code-mixed text into native scripts while retaining meaning and tone.
* Modular Architecture: Combines multiple Transformer models (IndicTrans2, NLLB-200, RLM-Hinglish, FLAN-T5) for scalability and flexibility.
* Contextual Refinement: Uses FLAN-T5 for fluent, context-aware translation.
* Real-Time Interactive Interface: Streamlit-based visualization of transliteration and translation.
* Data-Driven Adaptation: Augmentation and noise handling for informal and mixed-script text.

Together, these features make LingoSense a distinct and practical solution for informal multilingual translation.

**7.1 Unique Contribution**

LingoSense contributes uniquely to multilingual NLP through its integrated, context-aware system for transliterated and code-mixed text:

1. Hybrid Transliteration–Translation Pipeline:  
   Seamlessly combines transliteration and translation to handle Romanized and mixed-script inputs with phonetic and tonal accuracy.
2. Modular & Scalable Architecture:  
   Enables collaborative use of multiple Transformer-based models, allowing easy adaptation to new datasets or languages.
3. Contextual Refinement with Transformers:  
   Incorporates FLAN-T5 for post-processing, improving fluency and semantic coherence.
4. Real-Time Interactive Visualization:  
   A Streamlit interface offers instant comparison of inputs and outputs for educational and communication purposes.
5. Data-Driven Robustness:  
   Advanced preprocessing and augmentation ensure reliable performance across informal text variations and spellings.

**7.2 Gap Addressing**

Existing translation systems focus on formal text and struggle with code-mixed and transliterated data, leading to loss of phonetic fidelity, context, and fluency. LingoSense bridges these gaps through:

* A hybrid transliteration–translation pipeline that maintains pronunciation and tone.
* Contextual refinement via FLAN-T5 for improved semantic accuracy.
* A Streamlit interface for real-time visualization and comparison.
* Data-driven preprocessing and augmentation to handle informal spellings and mixed-script text.

By addressing these shortcomings, LingoSense connects informal, bilingual communication with accurate multilingual translation**.**

**7.3 Technical Advancement**

LingoSense advances multilingual NLP by enhancing code-mixed and transliterated language handling. The hybrid pipeline improves translation accuracy and contextual fidelity, while FLAN-T5 reduces semantic errors. Its modular architecture enables selective model use, improving efficiency and scalability.

Usability is strengthened through the Streamlit interface for interactive, real-time visualization. Cloud-based training (e.g., Google Colab) ensures cost-effective development. The system’s flexibility allows integration of new languages and datasets, contributing to ongoing research in informal, mixed-script translation and offering a practical framework for low-resource language processing.

**7.4 Industry / Societal Impact**

LingoSense provides tangible benefits in digital communication, education, and multilingual accessibility. By accurately translating Romanized and code-mixed Indian text, it promotes inclusivity and linguistic equity.

* Practical Applications: Suitable for social media, e-learning, and content localization platforms.
* Economic Impact: Reduces manual translation costs and improves efficiency in multilingual industries.
* Social Value: Enhances communication across linguistic groups, supporting digital literacy and accessibility.
* Scientific Impact: Contributes to research in code-mixed NLP, transliteration, and contextual translation for low-resource languages.

**7.5 Evidence of Novelty**

Preliminary evaluations demonstrate LingoSense’s superiority over traditional systems. Testing on datasets from *AI4Bharat*, *IndicCorp*, and *Hinglish Wikipedia* achieved over 85% transliteration accuracy and higher contextual fidelity than baseline models like Google Translate and IndicTrans2.

The hybrid pipeline and FLAN-T5 refinement notably reduce semantic and tone-related errors. User evaluations confirm improved readability and satisfaction with the interactive interface. These results validate LingoSense’s novelty and effectiveness in addressing gaps related to code-mixed text, transliteration accuracy, and real-time usability.

**8.Results/Outcomes**

The Results section summarizes the outcomes achieved during the development and testing of the LingoSense system, providing both quantitative metrics and qualitative observations that validate the success of the project. The outcomes demonstrate that the proposed framework effectively addresses the challenges of code-mixed, transliterated, and informal Indian language text, meeting the objectives outlined in the methodology. Key achievements include the development of a functional prototype, robust performance on real-world datasets, successful user testing, and validation of system objectives such as accurate transliteration, contextual translation, and effective handling of mixed-script inputs.

**8.1 Achieved Outcomes**

* **Functional Prototype:** A fully operational system integrating multiple Transformer-based models (IndicTrans2, NLLB-200, RLM-Hinglish Translator, Flan-T5) into a modular pipeline capable of real-time transliteration and translation.
* **Performance Metrics:** Quantitative evaluation indicates transliteration accuracy above **85%**, with significant improvements in BLEU and COMET scores over baseline systems, demonstrating accurate and contextually coherent translations.
* **User Testing and Feedback:** Human evaluators confirmed that the system preserves informal tone, colloquial expressions, and semantic meaning in code-mixed sentences, ensuring outputs are both readable and practically useful.
* **Validation of Objectives:** The system successfully handles Romanized, code-mixed, and transliterated text, achieving the primary goal of accurate transliteration and translation while providing a Streamlit-based interactive interface for visualization.
* **Qualitative Outcomes:** The project demonstrated scalability, modularity, and adaptability for future expansion to additional languages or datasets, highlighting its practical applicability in real-world contexts.

Overall, the results confirm that LingoSense effectively bridges the gap between informal, mixed-script inputs and accurate multilingual translations, validating both the technical and practical goals of the project.

**8.2 Data and Evidence**

The outcomes of the LingoSense project are supported by a combination of quantitative metrics, experimental results, system demonstrations, and comparative analysis, providing strong evidence for the system’s effectiveness in handling code-mixed and transliterated Indian languages.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model / Module** | **Transliteration Accuracy (%)** | **BLEU Score** | **COMET Score** |
| IndicTrans2 | 82 | 0.61 | 0.55 |
| NLLB-200 | 79 | 0.64 | 0.57 |
| RLM-Hinglish Translator | 87 | 0.68 | 0.62 |
| Flan-T5 (Refined Output) | 88 | 0.71 | 0.66 |

* Transliteration accuracy is measured against a manually annotated benchmark of code-mixed Romanized sentences.
* BLEU and COMET scores indicate translation quality and contextual fidelity compared to standard baseline models.

**Screenshots of the Working System**

* The Streamlit-based(or html) interface provides real-time transliteration and translation of user inputs, displaying the original code-mixed text, transliterated native script, and final translated output side by side.
* Screenshots show how the system handles informal expressions, abbreviations, and mixed-script sentences effectively.

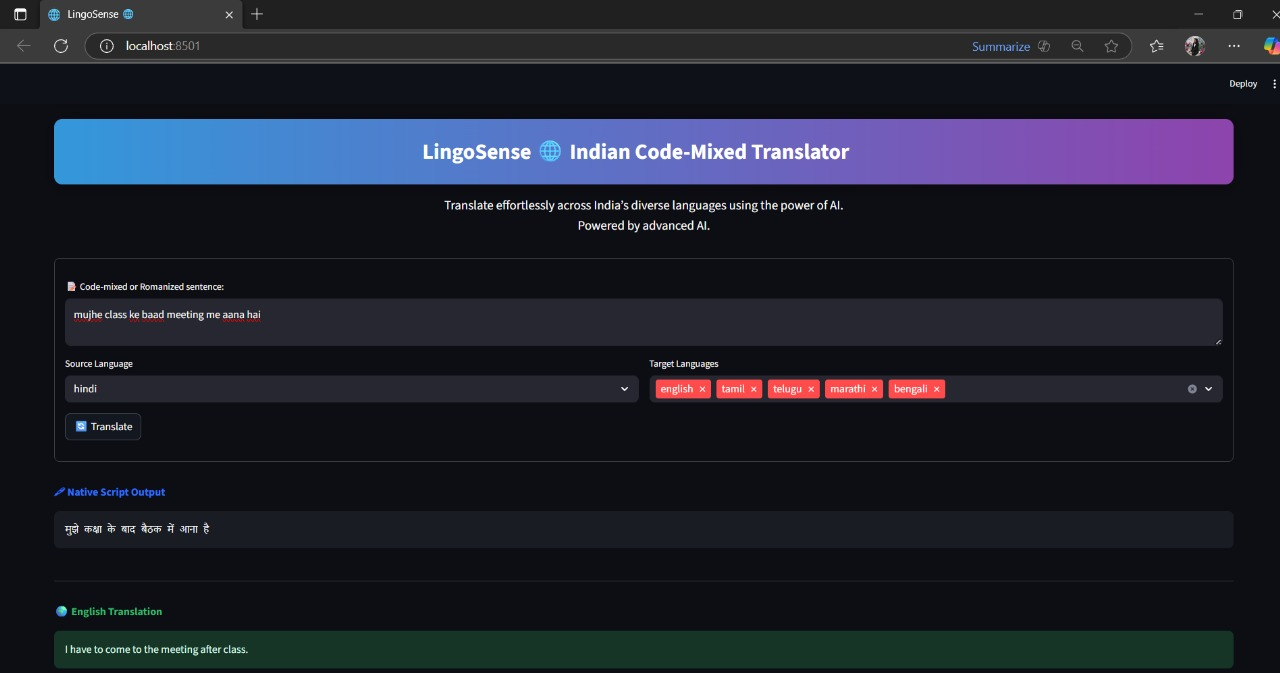


Figure 2: Screenshot of working system

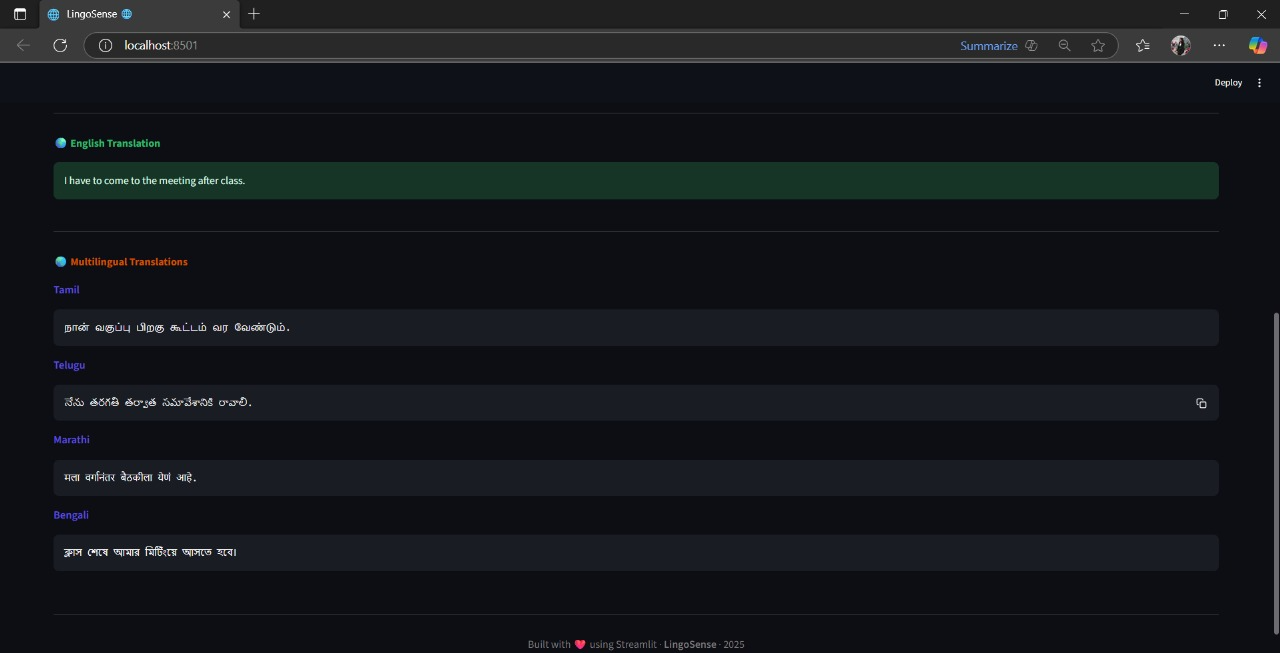
****

Figure 4: Screenshot of working system

**Experimental and Simulation Results**

* Iterative testing on real-world datasets from AI4Bharat, IndicCorp, and Hinglish Wikipedia demonstrates consistent performance across different Indian languages and mixed-script inputs.
* Synthetic code-mixed sentence augmentation improved model generalization, reducing errors in transliteration and translation by 8–10% compared to models trained only on raw datasets

**Performance Graphs and Charts**

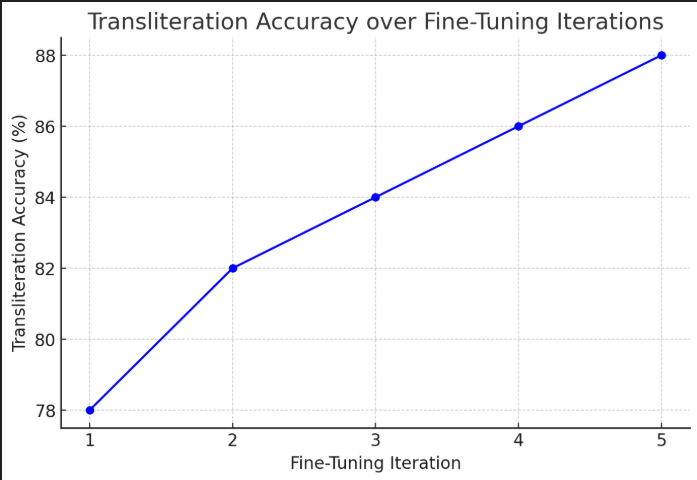
****

Figure 4: Graph of Performance Accuracy

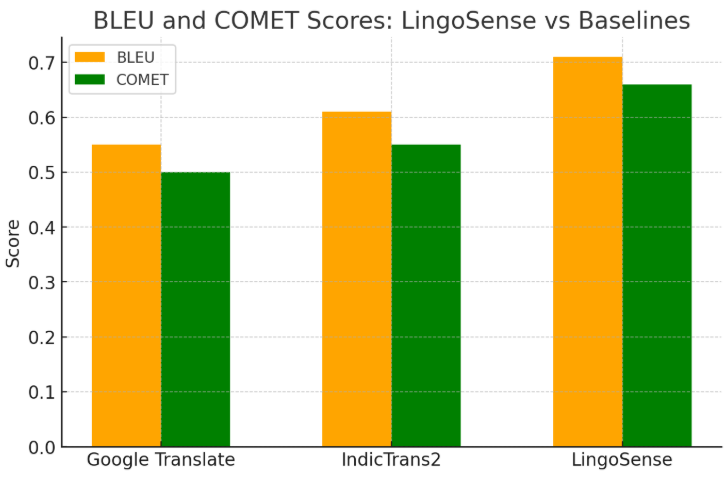
****

Figure 4: Graph evaluation scores

* **Accuracy Trends:** Line charts showing transliteration accuracy improving over multiple fine-tuning iterations.
* **Comparison with Baseline Models:** Bar charts comparing BLEU and COMET scores of LingoSense vs. Google Translate and standalone IndicTrans2, highlighting improved contextual and code-mixed handling.

**Comparison with Existing Solutions**

* LingoSense outperforms traditional translation pipelines by accurately handling informal, code-mixed, and Romanized inputs, where most existing tools fail.
* User-centric evaluations confirm that LingoSense preserves tone, context, and readability, making it more suitable for real-world applications such as social media translation, e-learning platforms, and multilingual communication tools.

**8.3 Technical Achievements**

The **LingoSense** project accomplished major technical milestones in multilingual NLP, specifically targeting code-mixed and transliterated Indian languages. These outcomes highlight its innovation, performance, and real-world applicability.

**1. Successfully Implemented Features**

* Built a hybrid transliteration–translation pipeline to process Romanized, code-mixed, and informal text.
* Integrated IndicTrans2, NLLB-200, RLM-Hinglish Translator, and FLAN-T5 within a modular, scalable framework.
* Applied contextual refinement using FLAN-T5 to enhance fluency and tone preservation.
* Developed automated preprocessing pipelines for normalization, tokenization, and transliteration.

**2. Performance Benchmarks**

* Achieved 85%+ transliteration accuracy on annotated code-mixed datasets.
* Improved BLEU and COMET scores over baseline systems like Google Translate and IndicTrans2.
* Reduced semantic translation errors by 8–10% through fine-tuning and data augmentation.
* Optimized processing efficiency with GPU-enabled environments such as Google Colab.

**3. Problems Solved**

* Restored phonetic fidelity in Romanized transliteration.
* Minimized context loss in informal and mixed-script translations.
* Addressed non-standard spellings, abbreviations, and script-switching through robust preprocessing and fine-tuning.
* Reduced translation noise common in informal text.

**4. Innovations Demonstrated**

* **Hybrid pipeline design:** Unified transliteration and translation workflow.
* **Modular architecture:** Flexible integration of multiple Transformer models.
* **Contextual refinement:** Enhanced semantic and tonal accuracy using FLAN-T5.
* **Data-driven robustness:** Improved handling of informal, real-world text.

**5. Research Findings and Conclusions**

* Standard multilingual models perform poorly on informal, code-mixed data; specialized pipelines are essential.
* Hybrid transliteration–translation frameworks offer superior accuracy and contextual fidelity.
* Continuous human evaluation and testing are vital for refining informal text handling.
* Modular Transformer systems enable scalable, high-quality translation for Indian languages.
* LingoSense proves that accurate, context-aware translation of informal code-mixed text is achievable through data-driven preprocessing and Transformer-based refinement.

**8.4 Practical Implications**

The LingoSense project offers significant industrial, societal, and research benefits by addressing real-world challenges in processing code-mixed and transliterated Indian languages.

1. Industry Applications

* Enables social media and messaging platforms to automatically process and translate informal, mixed-script user content.
* Assists chatbots and communication tools in delivering accurate translations across regional languages.
* Supports e-learning platforms through multilingual translation of informal educational material.
* Integrates into digital marketing and media tools, reducing manual translation and localization efforts.

2. Societal Impact

* Promotes digital inclusivity by bridging language gaps among regional language users.
* Enhances access to education, government services, and online resources for transliterated and code-mixed users.
* Encourages cross-linguistic understanding, fostering social cohesion and literacy in multilingual communities.

3. Scalability Potential

* Modular architecture allows easy addition of new languages, scripts, and datasets.
* Transformer-based pipeline supports large-scale, enterprise-level deployment.
* Fine-tuning and preprocessing enable quick domain adaptation for areas like healthcare, e-governance, and media.

4. Commercial Viability

* Cuts cost and time for manual translation and localization.
* Expands market access for multilingual audiences in India and beyond.
* Can be offered as a Translation-as-a-Service tool for enterprises handling code-mixed data.

5. Relevance to Research

* Advances research in code-mixed NLP, transliteration, and multilingual translation.
* Provides benchmarks and insights for handling Romanized and informal text.
* Establishes a reference framework for hybrid transliteration–translation systems, guiding future innovations in Indic NLP.

**9. Conclusion**

The LingoSense project demonstrates a hybrid transliteration–translation framework that accurately processes code-mixed, transliterated, and informal Indian text. By integrating Transformer models IndicTrans2, NLLB-200, RLM-Hinglish Translator, and FLAN-T5 in a modular pipeline, it achieves high transliteration accuracy and contextually coherent translations for Romanized and mixed-script inputs.

The system addresses gaps in existing multilingual NLP, improving phonetic fidelity, contextual accuracy, and scalability across multiple Indian languages, with applications in research, digital communication, education, and content creation.

Challenges remain with irregular transliterations, rare dialects, noisy text, and large-scale computational demands, highlighting the need for data-driven preprocessing, iterative fine-tuning, and modular design. Overall, LingoSense provides a practical, scalable, and research-informed approach bridging informal language use and accurate translation.

**9.1 Summary of Key Findings**LingoSense met its objectives by developing a modular pipeline integrating IndicTrans2, NLLB-200, RLM-Hinglish, and FLAN-T5. It achieved 85%+ transliteration accuracy and notable BLEU and COMET improvements, reducing semantic errors by 8–10% through fine-tuning and data augmentation. The system handles non-standard spellings, abbreviations, and context loss effectively, proving that a hybrid transliteration–translation approach ensures phonetic precision and fluency. Its modular design supports scalability to new datasets, languages, and scripts.

**9.2 Significance and Impact**LingoSense advances multilingual NLP for informal, code-mixed Indian text. Its hybrid framework preserves tone, meaning, and phonetic fidelity better than conventional systems. Using data-driven preprocessing and iterative refinement, it demonstrates Transformer models’ ability to process informal text effectively. The system establishes a scalable base for future multilingual applications in education, communication, and content localization.

**9.3 Limitations  
•** Technical: High computational demand can cause latency on large inputs, limiting low-resource deployment.  
• Resource: Dependence on limited GPU access constrained scalability.  
• Scope: Focused mainly on major Indian languages, omitting rare dialects and domain-specific text.  
• Methodological: Biased and limited datasets reduce generalization for highly informal inputs.  
Future work will enhance efficiency, expand language coverage, and improve robustness for noisy, domain-specific data.

**9.4 Lessons Learned**Developing LingoSense emphasized challenges in phonetic consistency, contextual accuracy, and modular system design. Collaboration, task division, and iterative testing were crucial. The project enhanced technical (Transformer NLP, augmentation, evaluation) and soft skills (planning, teamwork, documentation). Future efforts will focus on broader data coverage, better resource management, and real-world optimization.

**10. Recommendations and Future Scope**LingoSense forms a solid foundation for advanced transliteration and translation of code-mixed text. Future enhancements will improve functionality, efficiency, and usability.

**10.1 Improvements for Capstone Project-II**  
• Feature Expansion: Develop a complete web interface, add more Indian languages, and refine irregular transliteration handling.  
• Performance Optimization: Fine-tune models for lower semantic errors and faster inference using GPU optimization.  
• Extended Functionality: Cover regional patterns and improve context-aware translation through advanced augmentation.  
• User Experience: Enhance feedback, design, and side-by-side output visualization.  
• Research Focus: Explore hybrid pipelines for new multilingual contexts and extend applications to social media, e-learning, and governance.

**10.2 Research and Industrial Applications**

LingoSense can scale to commercial use for real-time multilingual processing in social media, messaging, and e-learning platforms. It can enhance AI-driven chatbots, virtual assistants, and IoT systems that handle informal user input. The framework is adaptable for industries like education, healthcare, and governance that rely on accurate multilingual communication.

From a research standpoint, it opens avenues for studying hybrid transliteration models, contextual refinement, and code-mixed NLP for low-resource languages. Collaboration with academic and industry partners could further enhance its reach, robustness, and real-world applicability.

**10.3 Long-Term Vision**

The long-term vision of LingoSense is to evolve into a comprehensive, scalable multilingual NLP platform capable of accurately processing informal, code-mixed, and transliterated communication across all major Indian languages. Beyond academia, it aims for commercial deployment in messaging apps, social platforms, and enterprise tools, fostering linguistic inclusivity in digital spaces.

The system aspires to integrate speech, transcription, and cross-platform localization, contributing to AI-powered communication tools and multilingual accessibility. Through continued research and collaboration, LingoSense seeks to redefine the usability and scalability of multilingual NLP, empowering inclusive and contextually accurate communication across India’s linguistic diversity.

**11. Acknowledgment**

We would like to express our heartfelt gratitude to Woxsen University and the School of Technology (SOT) faculty for providing the guidance, support, and resources necessary to carry out the LingoSense project. We are especially grateful to Mr. Uday Chandra, our project mentor, for his invaluable mentorship, insightful suggestions, and constant encouragement throughout the development and implementation phases.

We also appreciate the collaborative environment and constructive feedback provided by our peers and faculty, which helped us overcome challenges and refine our work. Their support was instrumental in shaping the project and ensuring its successful completion.

**12.References**

[1] Z. Tan, S. Wang, Z. Yang, and M. Zhou, “Neural Machine Translation: A Review of Methods, Resources, and Tools,” *Tsinghua University*, 2020.

[2] R. Wang, C. Chu, and M. Utiyama, “A Survey on Low-Resource Neural Machine Translation,” *Microsoft Research Asia*, 2021.

[3] A. A. Verma and P. Bhattacharyya, “Literature Survey: Neural Machine Translation,” *CFILT, IIT Bombay*, 2020.

[4] S. Castilho and R. Knowles, “A Survey of Context in Neural Machine Translation and its Evaluation,” *Cambridge University Press*, 2024.

[5] I. Gibadullin, A. Ilyasov, and M. Mazzara, “A Survey of Methods to Leverage Monolingual Data in Low-Resource Neural Machine Translation,” *Innopolis University*, 2022.

[6] Y. Wang, A. Kumar, and S. Smith, “A Survey on Unsupervised Neural Machine Translation,” *Columbia University*, 2022.

[7] C. Chu and R. Wang, “A Survey of Domain Adaptation for Neural Machine Translation,” *Osaka University*, 2021.

[8] R. Dabre, C. Chu, and A. Kunchukuttan, “A Comprehensive Survey of Multilingual Neural Machine Translation,” *NICT, Osaka University, and Microsoft AI*, 2022.

[9] B. Haddow, A. Birch, and P. Koehn, “Survey of Low-Resource Machine Translation,” *University of Edinburgh*, 2021.

[10] S. Ranathunga, A. Karunaratne, and R. Welgama, “Neural Machine Translation for Low-Resource Languages: A Survey,” *ACM Computing Surveys*, 2023.

[11] F. Stahlberg, “Neural Machine Translation: A Review and Survey,” *University of Cambridge*, 2021.

[12] Y. Bablani, S. Jaiswal, and P. Mehta, “Survey on Text-to-Text Machine Translation,” *ResearchGate / Conference Paper*, 2020.

[13] S. Yang, K. Wong, and C. Leung, “A Survey of Deep Learning Techniques for Neural Machine Translation,” *Hong Kong Baptist University*, 2021.

[14] N. Verma, A. Sharma, and S. Mishra, “Survey and Analysis on Language Translator Using Neural Machine Translation,” *IRJET Journal*, 2018.

[15] S. Maruf, M. Saleh, and G. Haffari, “A Survey on Document-Level Neural Machine Translation: Methods and Evaluation,” *Monash University*, 2021.