Multilingual Model for TEXT PROCESSING

Presented by:

Harika(SE23UARI097),

Ananya(SE23UARI087)

Sneha (SE23UARI121)

Srija (SE23UARI094)

November 28, 2025

Project Proposal & Refinement

This proposal outlines our initial concept for a unified text-processing system, detailing its core components. The design has been iteratively refined based on extensive feedback to ensure comprehensive and effective solutions.

1

Transliteration

A foundational element of the proposal, designed to cover all 22 scheduled Indian languages, ensuring seamless script conversion.

2

Normalization

A crucial component, trained on 12 languages, focused on verbalizing numbers, dates, and expanding abbreviations for linguistic consistency.

3

Punctuation Restoration

An essential aspect of the system, trained across 23 languages, to accurately recover and place structural punctuation cues for enhanced readability.

This refined proposal incorporates valuable insights and adjustments gathered from initial concept reviews, strengthening our approach to achieving robust multilingual performance.

Problem Identification & Gap Analysis

India's linguistic diversity, with 22 official languages, presents significant challenges for developing inclusive digital content, effective search functionalities, and robust Natural Language Processing (NLP) technologies.

A critical gap exists in comprehensive, unified pre-processing for these languages, particularly concerning accurate **transliteration**, consistent **text normalization**, and reliable **punctuation restoration**. Without these foundational steps, existing NLP models struggle with data heterogeneity and context preservation.

Lack of Unified Pre-processing

The absence of a standardized approach across diverse scripts leads to inconsistent digital content and unreliable NLP performance for Indian languages.

Fragmented Linguistic Solutions

Current solutions for script conversion, text cleaning, and punctuation recovery are often disparate, requiring multiple passes and reducing overall efficiency.

Impact on Readability & Context

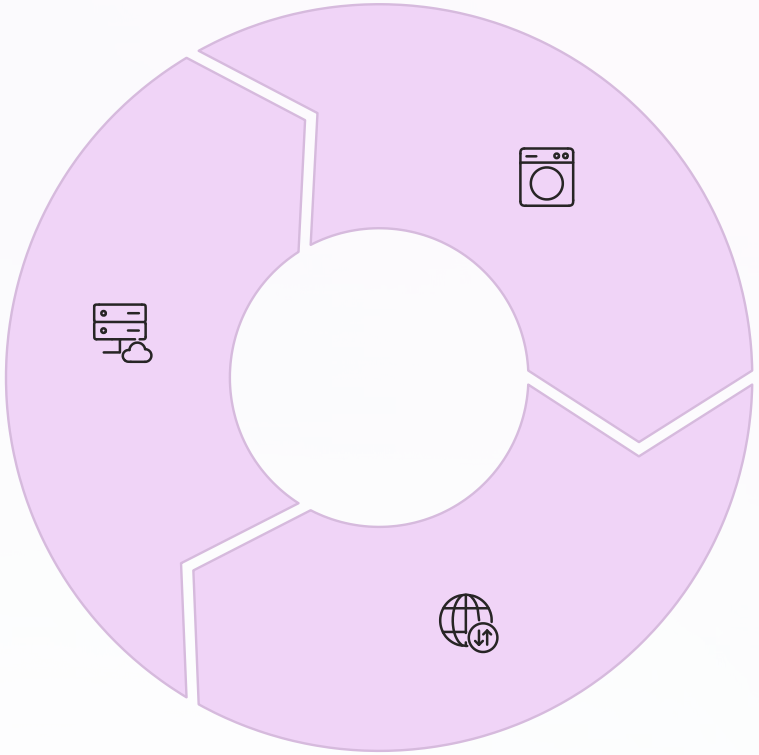
Poor punctuation and inconsistent text forms hinder readability, distort meaning, and pose significant barriers to accurate language understanding in NLP systems.

Methodology & Approach

Our comprehensive methodology leverages a unified multilingual text-processing system, built on Gemma-3-1B-PT, to address the challenges of linguistic diversity in digital content.

Curated Dataset

Development of high-quality, linguistically diverse datasets spanning various domains and scripts, essential for robust model training.



Rigorous Training Techniques

Application of multi-task fine-tuning, strong regularization, and robust validation methods to ensure model accuracy and generalization.

Multi-Task Processing Approach

Integrated processing for transliteration (22 languages), text normalization (12 languages), and punctuation restoration (23 languages) within a single system.

Datasets: Comprehensive & Curated

We constructed a comprehensive dataset for all 22 scheduled Indian languages, combining existing and newly transliterated corpora.

Existing Corpora

Updesh-beta: Multilingual QA and reasoning dataset.

IndicAlign: Pre-training and fine-tuning data.

Gemini-2.0-Flash

Sangraha, IndicCorp-v2, Cosmopedia-Translated, BPCC.

Click here

Datasets

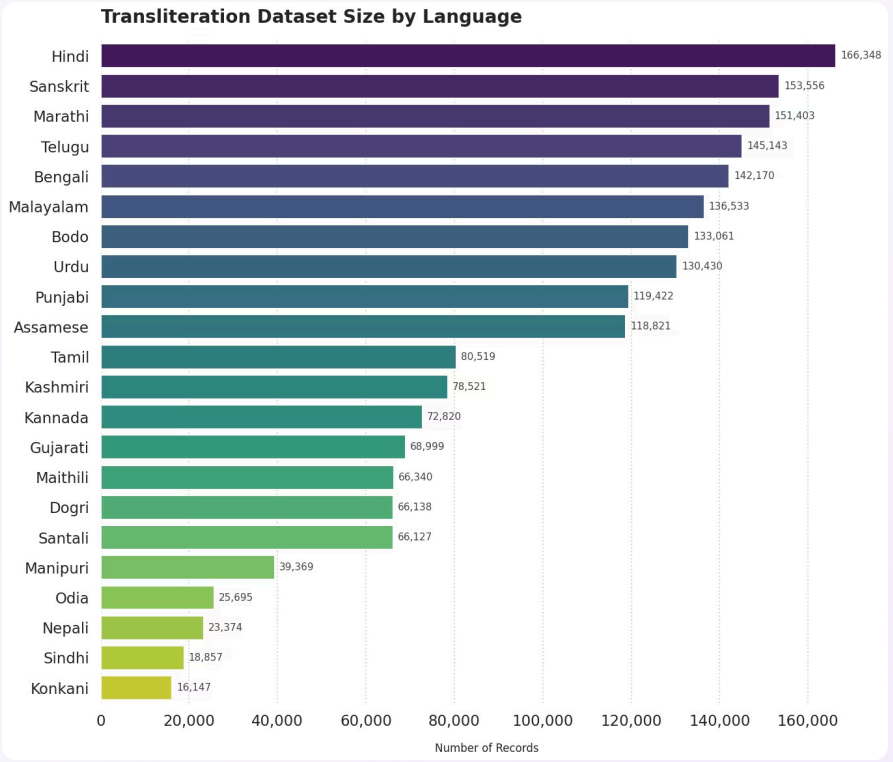
Transliteration

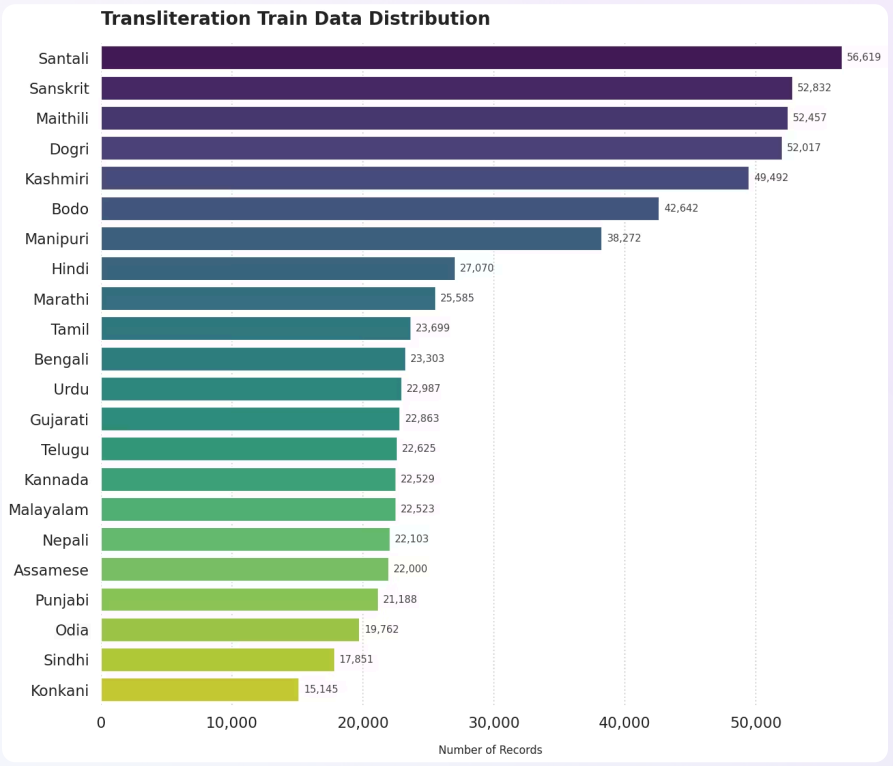
Existing Corpora with Native Transliteration

- Updesh-beta – Multilingual QA and reasoning dataset with native transliterations.
- IndicAlign – Part of IndicLMSuite; provides pre-training & fine-tuning data with native transliterations.

Corpora Transliterated Using Gemini-2.0-Flash

- Transliterated Sangraha – Derived from IndicLMSuite, transliterated using Gemini-2.0-Flash.
- IndicCorp-v2 Transliterated – Large-scale monolingual corpus transliterated using Gemini-2.0-Flash.
- Indic2Vec Transliterated – Multilingual pronunciation & speech corpus transliterated using Gemini-2.0-Flash.
- BPCC Transliterated – Bharat Parallel Corpus Collection transliterated using Gemini-2.0-Flash.





System Prompt for Synthetic Data Generation (Normalization Model)

Used to generate unnormalized Indian-English text for training a normalization model.

Ensures Indian context by including:

- Names (Priya Sharma, Rakesh Kumar)
- Places (Delhi, Mumbai, Kerala)
- Institutions (AIIMS, SBI, IIT Bombay)
- Cultural elements (Diwali, chai, Bollywood)

Output varies based on:

- Category (Medical, Legal, Tech, Travel, etc.)
- Subcategory
- Document Type (Article, Reddit post, Blog, etc.)
- Target Length
- Tone (formal, informal, conversational)

Requirements:

- Generated text must remain unnormalized (Dr., 15/08/2024, Rs. 500 etc.)
- Must be coherent, natural, and realistic.

Core Instructions & Examples

Core Instructions:

- Keep abbreviations, numerals, dates, and raw text as-is
- Integrate 3–8 unnormalized elements naturally
- Maintain coherence and human-like flow
- Match the tone and structure to the document type
- Length depends on type (e.g., Article: 300–500 words)

Click here to view the full system prompt

Click here

Examples:

Medical:

"Dr. Priya Sharma prescribed Paracetamol 500mg on 10/03/2023 at AIIMS Delhi."

Legal:

"Adv. Gupta filed a PIL in Bombay HC under IPC 302 on 15/07/2022."

Tech:

"Realme GT has 8GB RAM and launched on 10/10/2022."

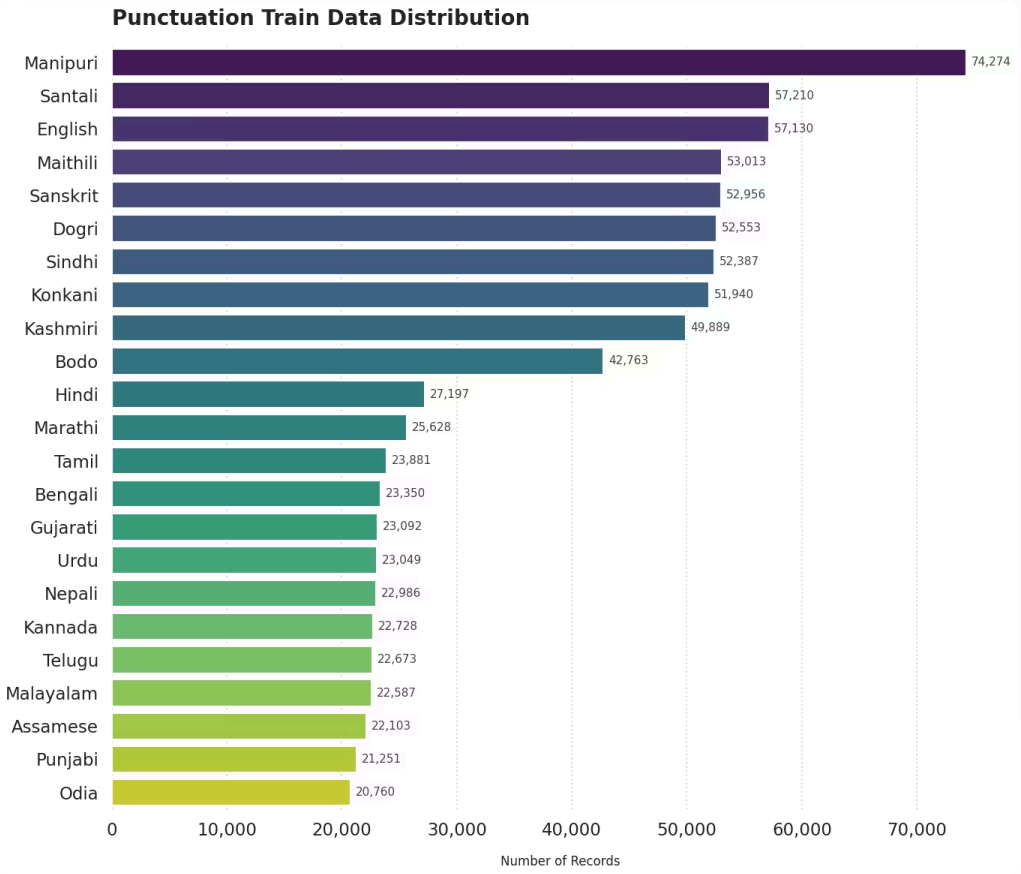
Currency:

"He exchanged $100 for ₹8300 at Mumbai airport."

Travel:

"Shatabdi departs Delhi 6:00 AM; reaches Jaipur at 10:30 AM."

Punctuation Restoration Dataset



Our punctuation restoration dataset covers 23 Indian languages with comprehensive training data distribution across diverse linguistic families.

Data Sources:

- IndicCorp v2
- BPCC (Bharat Parallel Corpus Collection)
- Sangraha
- IndicAlign

The dataset shows strong coverage across major Indian languages, with Manipuri, Santali, and English having the highest number of training records, ensuring robust model performance across linguistic diversity.

Methodology: Unified Problem Formulation & Implementation

All three tasks are cast as conditional generation, using prompt-conditioned inputs.

Transliteration

"Transliterate to Latin:" | x

Normalization

"Normalize numbers, dates, addresses; produce human-readable text:" | x

Punctuation

"Restore punctuation and casing faithfully:" | x

We fine-tune Gemma-3-1B-PT, a decoder-only Transformer with shared vocabulary across tasks.

Supervised Fine-Tuning on Gemma 3-1B-Pt

Objective

Adapt a pretrained Gemma 3-1B-Pt foundation model for high-accuracy multilingual text tasks.

Method

Supervised fine-tuning using aligned instruction response pairs across Indian scheduled languages.

Highlight

Fine-tuned model aligns task behavior with Indian language linguistic structure and orthography.

Training Strategy

- Strict, instruction following objectives
- High-quality parallel pairs for translation, transliteration, and normalization
- Token level supervision for consistency across scripts

Outcomes

- Strong cross-script generalization
- Significant improvement in accuracy, consistency, and fluency
- Reduced hallucinations and error-drift

Multilingual Text Normalization – System Overview

Text Normalization

1

Purpose of Normalization

- Convert unnormalized → human-readable text
- Handle abbreviations, numerals, dates, shorthand
- Produce consistent clean outputs

2

Languages Covered

- 12-language normalization set
- English as pivot for normalization
- Multilingual supervision generated by translating normalized English into other languages

3

Data Construction

- Build (unnormalized, normalized) parallel pairs
- Merge pairs to align original unnormalized text with normalized targets
- Ensure high variety: dates, numerals, currency, abbreviations, domain terms

4

Data Pipeline

- Normalize English first
- Translate normalized output to target languages

5

System Context

- Works alongside transliteration & punctuation restoration
- Uses high-quality curated datasets
- Gemma-3-1B-PT model backbone

Key Challenges Faced During the Evaluation Pipeline

1

Incomplete Output During Translation (Normalization Issue)

Problem: While translating both normalized and unnormalized samples using the Gemini API, we observed that the API occasionally truncated the output, especially towards the end of sentences.

Solution: We used Python's difflib to automatically compare the unnormalized reference with the truncated normalized output. The missing portion was identified and then appended back, ensuring the final normalized output was complete and accurate.

2

Meaning Drift Between Normalized & Unnormalized Translations

Problem: Since we translated normalized and unnormalized samples independently using the Gemini API, their outputs sometimes had slightly different word choices. Although the overall meaning stayed the same, the literal phrasing diverged.

Solution: We merged both translations intelligently so that the final output preserved the same meaning and avoided inconsistencies caused by separate translations.

3

Penalization Issue in Punctuation Restoration

Problem: For the punctuation restoration task (e.g., "hello" → "hello."), the model tended to assign very high probabilities to repeated tokens like "hello", which meant it did not penalize missing punctuation strongly enough.

Solution: We applied temperature tuning to increase the model's sensitivity to punctuation-related variations, ensuring better prediction diversity and proper penalization of incorrect outputs.

Evaluation Metrics

Comprehensive Assessment Framework for Transliteration Quality

1

BLEU

Bilingual Evaluation Understudy

Measures how similar the predicted transliteration is to the reference using matching n-grams.

Formula:

$$BLEU = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Where: *pn* = n-gram precision, *wn* = weights, *BP* = brevity penalty.

Interpretation:

- >0.80 Excellent
- 0.60-0.80 Good
- 0.40-0.60 Average
- <0.40 Poor

💡 Higher BLEU → Better performance

2

ChrF++

Character n-gram F-score

Evaluates matching character sequences, combining precision + recall using character-level n-grams.

Formula:

$$CHRF++ = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}$$

Where: β = weight parameter (usually 2).

Interpretation:

- >0.70 Excellent
- 0.50-0.70 Good
- <0.50 Needs improvement

💡 Captures both spelling accuracy and phonetic structure.

3

WER

Word Error Rate

Measures word-level errors between predicted and reference transliteration.

Formula:

$$WER = \frac{S + D + I}{N}$$

Where: *S* = substitutions, *D* = deletions, *I* = insertions, *N* = total words in reference.

Interpretation:

- 0.00 Perfect
- 0.00-0.20 Strong
- 0.20-0.40 Moderate
- >0.40 Weak

💡 Lower WER = Higher correctness

4

CER

Character Error Rate

Measures character-level mistakes, very strict but important for transliteration.

Formula:

$$CER = \frac{S + D + I}{N}$$

Where: *S* = substitutions, *D* = deletions, *I* = insertions, *N* = total characters in reference.

Interpretation:

- 0.00-0.10 Excellent
- 0.10-0.20 Good
- >0.20 Weak

💡 Detects even single-character phonetic/spelling mistakes.

Why all four metrics?

Metric	Measures	Focus
BLEU	n-gram match	Sentence-level accuracy
ChrF++	character n-gram match	Phonetic + Spelling accuracy
WER	word-level errors	Readability
CER	character-level errors	Precision

Together, these metrics provide complete evaluation of transliteration quality — global similarity, local character accuracy, readability, and error sensitivity.

Sarvam Model Performance — Baseline Metrics

Language-wise BLEU, ChrF++, WER, and CER Scores

Language	BLEU	ChrF++	WER	CER
Bengali	11.97	38.69	0.805	0.354
Gujarati	20.52	50.45	0.68	0.255
Hindi	37.55	65.19	0.473	0.324
Kannada	12.29	42.21	0.830	0.382
Malayalam	10.05	38.12	0.89	0.44
Marathi	15.93	49.36	0.699	0.247
Tamil	9.49	33.81	0.896	0.578
Telugu	11.24	40.01	0.880	0.426

sers > harik > Desktop > test > language_wise_metrics.csv > data

```
Language,Count,BLEU,ChrF++,WER,CER
Bengali,1000,11.974442400867183,38.69645586820886,0.8053137568024853,0.3543088334680832
Gujarati,860,20.522550186530086,50.452634220863125,0.6861938585320981,0.2547977273592907
Hindi,1000,37.55116748235164,65.18519065068153,0.4727696297981787,0.3237340142467221
Kannada,998,12.291082393442817,42.21440089797109,0.829985741773472,0.38256500631088663
Malayalam,1000,10.057390597428014,38.1238387537167,0.9464588713441369,0.44017395347354377
Marathi,996,15.938734231125451,49.35576965944083,0.6992990164092066,0.24685814121878766
Tamil,999,9.499824277658492,33.81474697879529,0.9957024486392839,0.5783848985358957
Telugu,998,11.243483043893432,40.005372557488144,0.8802687722814646,0.4256981962282844
```

Baseline performance metrics from Sarvam model evaluation across 8 Indian languages

Unified Model Performance — Our Results

Language-wise BLEU, ChrF++, WER, and CER Scores

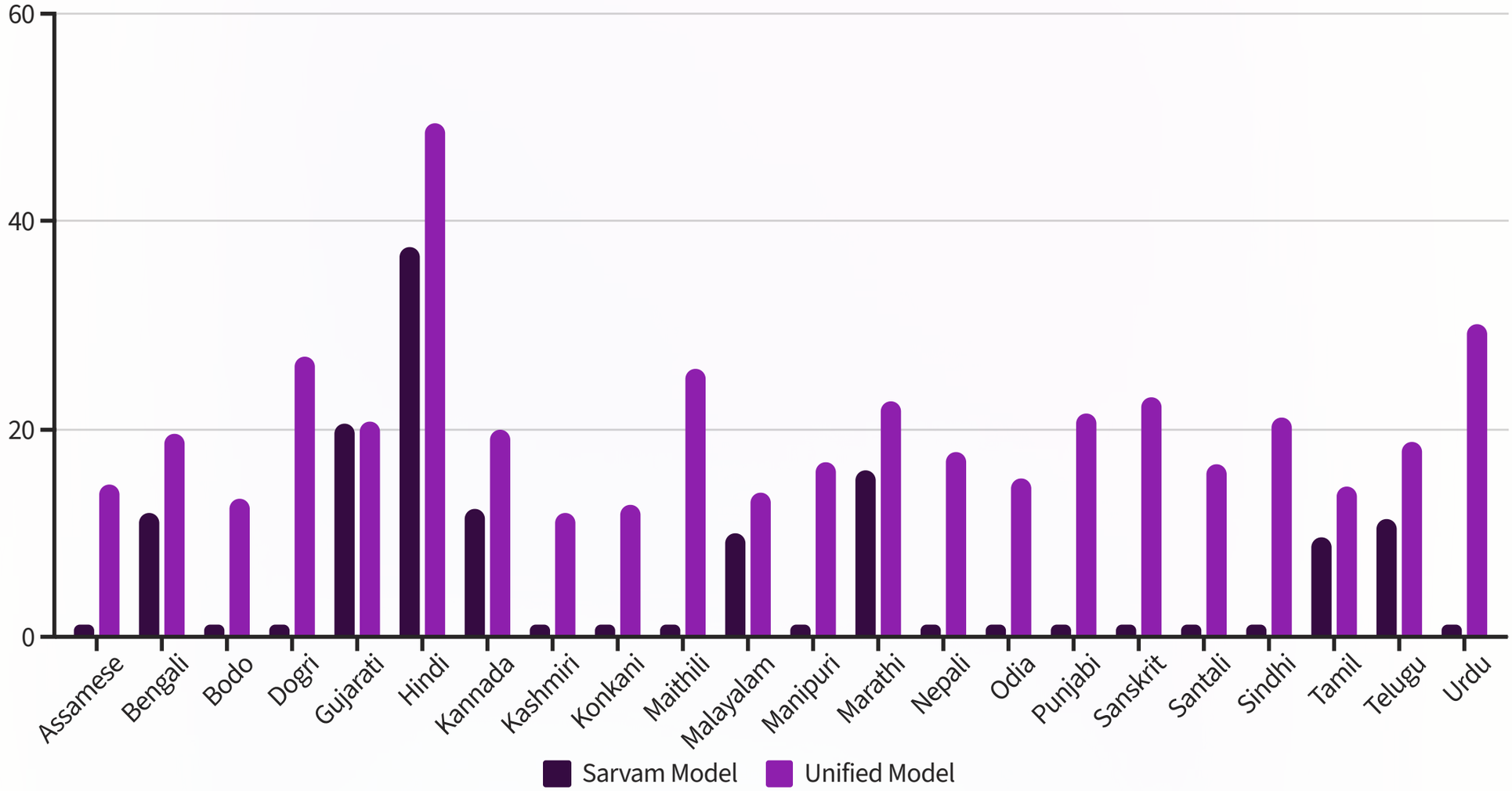
Language	BLEU	ChrF++	WER	CER
Assamese	14.71	42.84	0.714	0.293
Bengali	19.59	46.64	0.655	0.289
Bodo	13.38	50.71	0.69	0.213
Dogri	27.00	57.83	0.475	0.179
Gujarati	20.72	49.81	0.647	0.251
Hindi	49.47	73.25	0.337	0.168
Kannada	19.84	55.40	0.645	0.228
Kashmiri	11.92	42.36	0.696	0.276
Konkani	12.72	44.99	0.717	0.253
Maithili	25.82	60.55	0.539	0.163
Malayalam	13.79	44.56	0.771	0.321
Manipuri	16.81	47.12	0.697	0.285
Marathi	22.65	53.72	0.601	0.238
Nepali	17.71	54.18	0.62	0.196
Odia	15.21	51.34	0.67	0.215
Punjabi	21.48	48.98	0.595	0.258
Sanskrit	23.05	57.00	0.587	0.204
Santali	16.59	48.13	0.614	0.220
Sindhi	21.07	49.67	0.571	0.219
Tamil	14.47	45.37	0.743	0.317
Telugu	18.86	51.59	0.676	0.256
Urdu	30.13	57.95	0.486	0.210


```
Language,Count,BLEU,ChrF++,WER,CER
Assamese,679,14.715416996998565,42.843024958810055,0.7138830704943333,0.2932784737501824
Bengali,997,19.598768977173226,46.6449247761376,0.6548426906489832,0.289336939096798
Bodo,999,13.383236988657467,50.705811789461876,0.6999280422532849,0.2126675166627543
Dogri,1000,26.993861386741003,57.83542201210707,0.4753620072883696,0.17917137334080013
Gujarati,996,20.724659954724416,49.81453885703035,0.6468931937371346,0.25069909978630467
Hindi,1000,49.46632829385703,73.25185742441187,0.3366546175787 7185,0.1682490858269996
Kannada,995,19.83769347504502,55.40803102094202,0.6447208692206844,0.2277921929104576
Kashmiri,1000,11.915469027727458,42.361082925470484,0.6955121774277817,0.2760104556747758
Konkani,1000,12.723719988601777,44.99145147896692,0.7166893345081198,0.25288356800880785
Maithili,1000,25.82270607858087,60.55827224265753,0.530551440173256,0.162591329056741 73
Malayalam,994,13.788807145196419,44.56076701330738,0.7707888123492719,0.320695310666787 47
Manipuri,999,16.807111511750342,47.122587994448104,0.6968054984172406,0.28497796848189355
Marathi,998,22.650536245496813,53.72750523438511,0.6006238011081345,0.23814802092765983
Nepali,999,17.708947404207574,54.18907197799807,0.6292522002397474,0.1964309313509109
Odia,997,15.208110441002363,51.34196562333267,0.6797548594746599,0.21491132147707295
Punjabi,996,21.478045995981997,48.98712258479951,0.594725364222791,0.25707008104919854
Sanskrit,1000,23.054423576168645,57.004018691401384,0.5873198112218008,0.203372045836160 06
Santali,999,16.585451350853397,48.13777863056757,0.6138130445257626,0.21997721196008882
Sindhi,1000,21.06921923521979,49.67584925894167,0.5719563695701739,0.23889144475983262
Tamil,986,14.4681991178483 98,45.375985098499726,0.7429166352914299,0.31659135427239943
Telugu,994,18.857148951913754,51.59585096310324,0.6758410678028447,0.2558956458019023
Urdu,998,30.13017804542542,57.95376415738458,0.4856252222932529 4,0.21010834972896822
```


Our Unified Model performance metrics across 22 Indian languages

BLEU Score Comparison: Sarvam vs Unified Model

Language-wise Performance Analysis



Our Unified Model shows consistent improvements across all languages, with significant gains in Hindi (+11.92), Bengali (+7.62), and Marathi (+6.72).

Evaluation Metrics & Validation

We evaluate the unified system across all tasks using consistent splits and clear metrics.

Metrics Used:

- **Transliteration (22 langs):** Character Error Rate (CER), Word Error Rate (WER), BLEU.
- **Normalization (12 langs):** CER/WER, slot-level exact match.
- **Punctuation Restoration (23 langs):** Token-level Precision, Recall, F1, BLEU.

These metrics capture accuracy at character and word levels, n-gram fidelity, and structural correctness.

References

Key resources that informed our research and dataset construction:

- Doddapaneni et al. (2023). IndicCorp v2: Monolingual corpora, benchmark and models for indic languages. **Hugging Face**.
- Gala et al. (2023). BPCC: Bharat Parallel Corpus Collection. **Hugging Face**.
- Khan et al. (2024). IndicLLMSuite: A blueprint for creating pre-training and fine-tuning datasets for indian languages. *arXiv preprint arXiv: 2403.06350*.
- Pulipaka et al. (2025). Mark my words: A robust multilingual model for punctuation in text and speech transcripts.
- Microsoft Research (2025). Updesh_beta: A multilingual question-answering and reasoning dataset. **Hugging Face**.