Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260501054
Volume 5, Issue1 (January-April 2026), PP: 478-491.
www.indjcst.com
ISSN No: 2583-5300

Advanced Multilingual Chatbot for Indian Language Support

Dr. Avinash. S. Kapse[1], Vaishali Datta Parihar[2]
[1,2]Department of Computer Science and Engineering, Anuradha Collage of Engineering & Technology, Chikhli, Maharashtra, India.

***To Cite this Article**: Dr. Avinash. S. Kapse[1], Vaishali Datta Parihar[2], "Advanced Multilingual Chatbot for Indian Language Support", Indian Journal of Computer Science and Technology, Volume 05, Issue 01 (January-April 2026), PP: 478-491.*

Abstract: *India's digital economy is expanding rapidly, yet over ninety percent of automated customer support systems remain accessible only in English, excluding the majority of users who prefer native-language interaction. This paper presents the design, implementation, and empirical evaluation of an intelligent multilingual chatbot platform supporting six major Indian languages — Hindi, Bengali, Marathi, Tamil, Telugu, and Kannada — spanning both the Indo-Aryan and Dravidian language families. The system is built on a microservices architecture integrating FastText-based language identification, fine-tuned multilingual BERT (mBERT) and IndicBERT models for intent classification and named entity recognition, a Redis-backed context-aware multi-turn dialogue engine, AI4Bharat IndicConformer for automatic speech recognition (ASR), and Bhashini and Sarvam AI APIs for text-to-speech synthesis (TTS). An annotated e-commerce dataset of 18,550 examples across six languages and twenty intent categories was constructed. Three core experiments were conducted: (1) a comparative evaluation of mBERT versus IndicBERT for intent classification, where per-language IndicBERT models achieved a macro-averaged F1-score of 89.2%, outperforming mBERT (82.6%) by 6.6 percentage points; (2) a context-aware dialogue ablation study demonstrating a 7.8% improvement in multi-turn accuracy over the single-turn baseline; and (3) an ASR benchmark where IndicConformer achieved an average Word Error Rate of 13.4% across six languages, outperforming fine-tuned Whisper on Hindi (11.2% vs 13.8% WER). User acceptance testing with 25 participants yielded a CSAT score of 4.2/5.0 and an 84% task completion rate.*

Key Words: *Multilingual NLP, Indian Languages, Chatbot, Intent Classification, IndicBERT, mBERT, ASR, IndicConformer, Dialogue Management, E-Commerce.*

I. INTRODUCTION

A. Background and Motivation

India is one of the world's most linguistically diverse nations, with twenty-two officially recognized languages and hundreds of regional dialects [1]. Despite this diversity, most automated customer support systems are designed exclusively for English-speaking users. Over ninety percent of Indian internet users prefer to communicate in their native language [1], yet high-value e-commerce actions such as tracking orders, initiating returns, and resolving payment disputes are routinely abandoned by non-English-speaking users who cannot communicate their needs effectively.

Transformer-based language models [9] such as BERT [3] and Indic-specific variants such as IndicBERT [4] and Indic Conformer [6] provide a strong technical foundation. However, their integration into complete, production-ready, multi- modal chatbot systems covering both Indo-Aryan and Dravidian language families simultaneously remains largely unexplored.

The Indian e-commerce market, valued at over $80 billion in 2024, serves a user base that is increasingly non-English and mobile-first [1]. Providing automated customer support in native languages is not merely an accessibility improvement — it is a business imperative with direct impact on user retention and customer satisfaction.

B. Problem Statement

Existing platforms such as Dialogflow CX, RASA [10], and Amazon Lex offer limited or poor-quality support for Indian languages. None provides an open-source, end-to-end framework combining multilingual Indian language text understanding, context-aware dialogue management, and voice interaction in a single integrated platform [4]. Furthermore, no published work provides empirical benchmarks comparing state-of-the-art multilingual models for Indian language customer support across both Indo-Aryan and Dravidian families.

C. Research Objectives

The following objectives guided this work:

1. Design and implement an end-to-end multilingual chatbot for six Indian languages across both Indo-Aryan and Dravidian families.
2. Empirically compare mBERT [3] versus IndicBERT [4] for e-commerce intent classification across all six languages.
3. Implement and evaluate a context-aware multi-turn dialogue engine using Redis session state [5].
4. Benchmark IndicConformer [6] against fine-tuned Whisper [8] for ASR across six languages on the IndicVoices dataset.

5. Validate the full system through user acceptance testing.

D. Research Contributions

This paper makes the following contributions:

- A complete open-source multilingual chatbot pipeline covering all stages from language detection to voice output for six Indian languages.
- The first empirical comparison of mBERT [3], IndicBERT (single multilingual) [4], and IndicBERT (per-language) for customer support intent classification across both language families, yielding a best macro F1-score of 89.2%.
- A multi-turn context retention architecture with a structured ablation study demonstrating 7.8% accuracy improvement over single-turn baselines.
- An ASR benchmarking study comparing IndicConformer [6] and fine-tuned Whisper [8] on six Indian languages, with an average WER of 13.4%.
- An annotated e-commerce intent dataset of 18,550 examples across six Indian languages, publicly released for research use.

E. Paper Organization

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system architecture and methodology. Section IV details the dataset and experimental setup. Section V presents and discusses experimental results. Section VI concludes the paper and outlines future work.

II.LITERATURE REVIEW

A. Multilingual Pre-trained Language Models

Vaswani et al. [9] introduced the Transformer architecture, which forms the foundation of all modern pre-trained language models. Devlin et al. [3] introduced BERT, demonstrating that a Transformer model pre-trained on large corpora could be fine-tuned for diverse NLP tasks. Multilingual BERT (mBERT), pre-trained on 104 Wikipedia corpora, extended this capability across languages but underperforms on morphologically rich and lower-resource languages due to unequal training data representation [3]. Kakwani et al. [4] released IndicBERT, pre-trained exclusively on eleven Indian languages using the AI4Bharat IndicCorp corpus, demonstrating consistently superior performance on Indic NLP benchmarks including IndicGLUE. Khanuja et al. [11] introduced MuRIL, which incorporates transliterated text alongside native scripts, improving code-mixed text understanding. Dabre et al. [12] released IndicBART, a sequence-to-sequence model suited for generative NLP tasks in Indian languages.

B. Chatbot Architectures and Dialogue Management

RASA [10] is a widely adopted open-source chatbot framework supporting custom NLU pipelines and story-based dialogue management, but it provides limited Indic script support. Commercial platforms such as Dialogflow CX and Amazon Lex offer stronger multilingual coverage but are proprietary and do not support quality Indian language text processing or voice [4]. Williams and Young [5] formalized dialogue state tracking using partially observable Markov decision processes (POMDPs), which informs the state machine approach adopted in this work. No published system covers more than two or three Indian languages simultaneously with context-aware multi-turn capability and voice integration.

C. Indian Language ASR

Radford et al. [8] introduced Whisper, an encoder-decoder ASR model trained on 680,000 hours of multilingual audio with strong zero-shot performance across 99 languages. However, Whisper's training data is heavily skewed toward English and European languages, resulting in higher error rates for Dravidian languages [8]. AI4Bharat released IndicConformer [6], a conformer-based ASR model purpose-built for 22 Indian languages using the IndicVoices dataset comprising over 7,000 hours of speech from 16,200+ speakers.

D. Code-Mixing in Indian Languages

Bhat et al. [13] demonstrated significant degradation of monolingual NLP models on code-mixed input and proposed word-level language tagging as mitigation. Code-mixing is pervasive in Indian digital communication, with studies reporting that over 40% of social media posts by Indian users contain some form of language mixing [13]. The proposed system handles code-mixing at the language detection stage using FastText's word-level confidence scores [2].

E. Gap Summary

Gap in Existing Literature	This Paper's Contribution
No mBERT vs. IndicBERT comparison for 6 Indian languages in customer support [3][4]	Experiments A, B, C (Section V-B)
No open-source context-aware multi-turn chatbot for Indian languages [5][10]	Redis state machine + entity carryover (Section III-C)
No end-to-end voice dialogue system for 6 Indian languages with WER benchmarks [6][8]	IndicConformer + Whisper comparison (Section V-E)
No annotated e-commerce intent dataset for 6 Indian languages [14][15]	18,550-example dataset (Section IV)

Table I Literature Gap Summary

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall System Architecture

The system adopts a five-layer microservices architecture in which each functional component operates as an independent, containerized service communicating via REST APIs. The five layers are: (1) Presentation Layer — a JavaScript-based embeddable chat widget and a React-based admin dashboard; (2) API Gateway Layer — a FastAPI-based gateway handling JWT authentication and rate limiting; (3) NLP Processing Layer — the core six-stage sequential pipeline; (4) Data Layer — PostgreSQL, MongoDB, and Redis with 30-minute TTL [5]; (5) Integration Layer — outbound connections to Bhashini [7], IndicTrans2, and Sarvam AI.

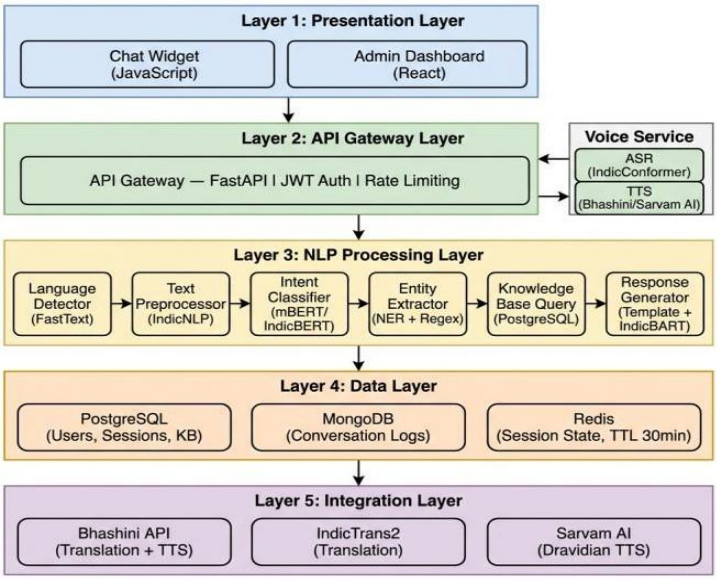

Fig. 1. Five-layer microservices architecture of the proposed multilingual chatbot platform.

B. NLP Processing Pipeline

Every user message passes through six sequential processing stages:

Stage 1 — Language Detection using FastText LID ; Stage 2 — Text Preprocessing using IndicNLP Library [16]; Stage 3 — Intent Classification using fine-tuned mBERT or IndicBERT [4]; Stage 4 — Entity Extraction using hybrid mBERT NER + regex [3]; Stage 5 — Knowledge Base Query from PostgreSQL with IndicBART [12] fallback; Stage 6 — Response Generation with entity slot filling and translation via Bhashini [7].

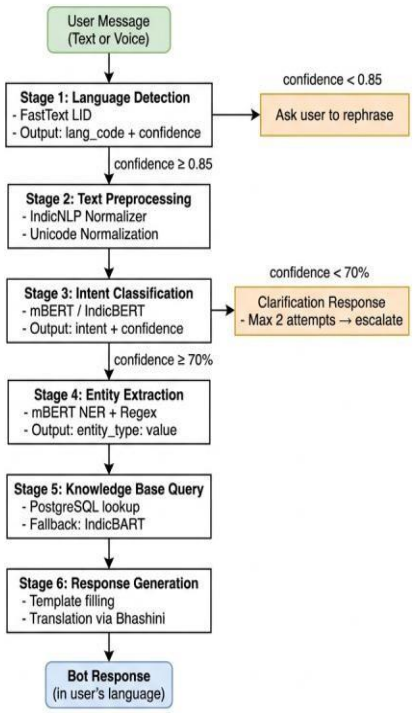

Fig. 2. Six-stage NLP processing pipeline. Every user message passes through all stages sequentially.

C. Context-Aware Dialogue Management

Phase 2 introduced context-aware multi-turn dialogue using a Redis-backed session state machine [5]. Each active session maintains entity carryover (most recent value overwrites), dialogue flow stage tracking via finite state machine, and a context window of the last 5 turns prepended as [CLS] current_message [SEP] turn_N-1 ... [SEP] turn_N-5 [3][4]. Five complex dialogue flows were implemented: return_product, cancel_order, change_address, exchange_product, and complaint_register. Session TTL is 30 minutes.

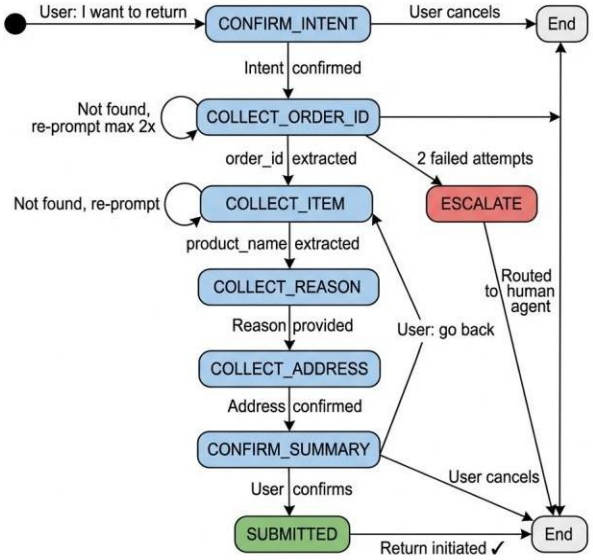

Fig. 3. Dialogue state machine for the return_product intent.

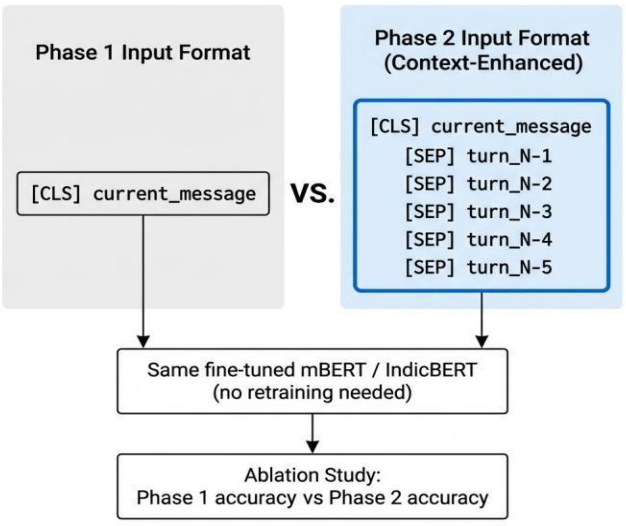

Fig. 4. Context injection approach — Phase 1 input format vs Phase 2 context-enhanced input format.

D. Voice Processing Pipeline

Phase 3 added voice I/O as a wrapper around the existing text pipeline. ASR: AI4Bharat IndicConformer [6] (primary, all 6 languages) and Whisper-medium [8] (fine-tuned, Hindi comparison). Audio preprocessed to 16kHz mono with noise reduction and silence trimming. TTS: Bhashini [7] for Indo-Aryan, Sarvam AI for Dravidian, Google Cloud TTS [19] as fallback. Output compressed using Opus codec.

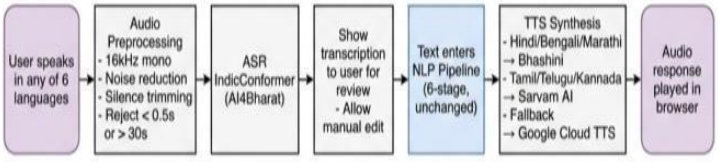

Fig. 5. Voice processing pipeline. ASR and TTS wrap the existing text pipeline without modifying it.

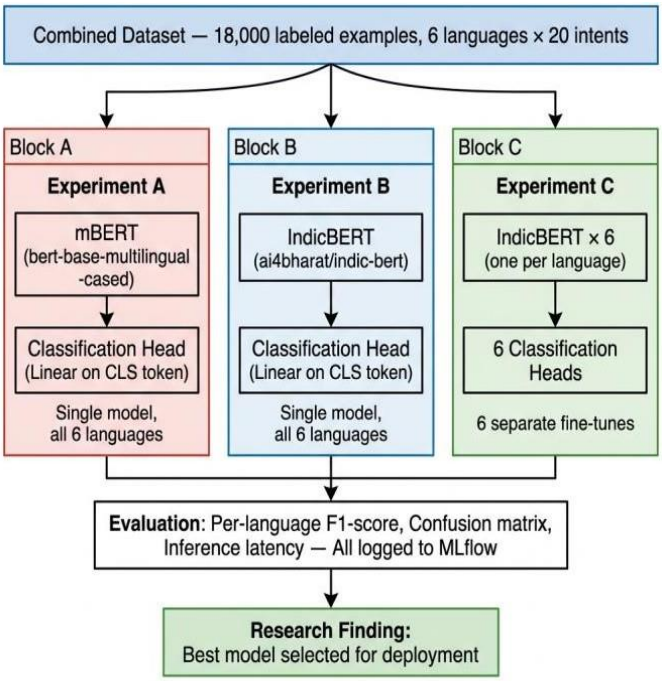

Fig. 6. TTS provider routing logic. Language determines the primary provider.

E. Intent Classification — Three-Experiment Design

Three model configurations were evaluated: Experiment A — mBERT (bert-base-multilingual-cased) [3] unified; Experiment B — IndicBERT (ai4bharat/indic-bert) [4] unified; Experiment C — IndicBERT × 6 (per-language) [4]. Standardized hyperparameters: lr=2e-5 with linear warmup, batch 32, 10 epochs, AdamW with weight decay 0.01, class- weighted cross-entropy loss [3]. All runs logged to MLflow.

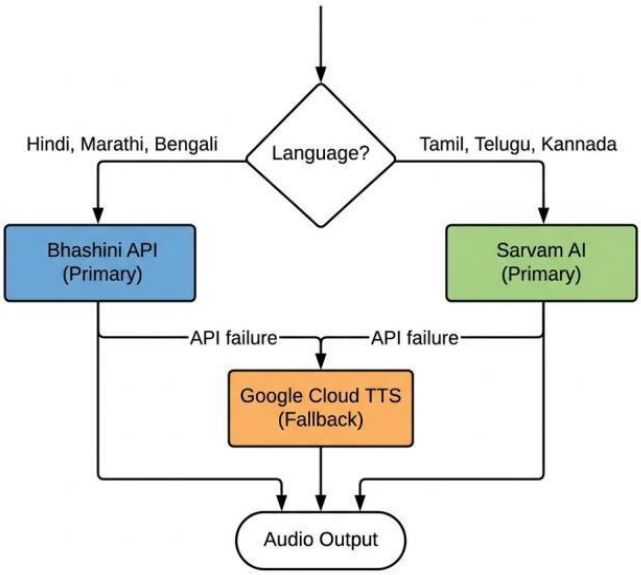

Fig. 7. Three-experiment intent classification comparison design.

IV. DATASET AND EXPERIMENTAL SETUP

A. Intent Taxonomy and Language Coverage

Six Indian languages are supported in priority order [4][14]: Hindi, Bengali, Marathi (Indo-Aryan) and Tamil, Telugu, Kannada (Dravidian). Twenty intent categories were defined for the e-commerce domain: track_order, cancel_order, return_product, refund_status, payment_issue, product_availability, shipping_cost, delivery_time, change_address, exchange_product, coupon_apply, account_login, order_modify, store_hours, complaint_register, product_review, warranty_info, bulk_order, gift_wrapping, and human_escalate. Ten entity types were defined.

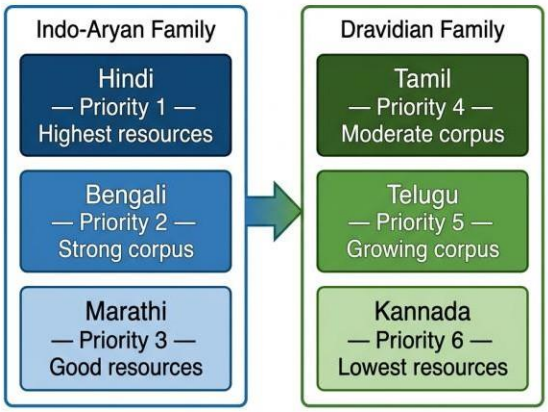

Fig. 8. Language priority order based on available corpus size and resource maturity.

B. Data Collection and Statistics

The dataset was constructed using four strategies: translation-based via Bhashini [7] with native speaker review (70%), synthetic generation via GPT-4 (15%), social media collection for code-mixed examples (10%), and crowdsourced native speakers (5%).

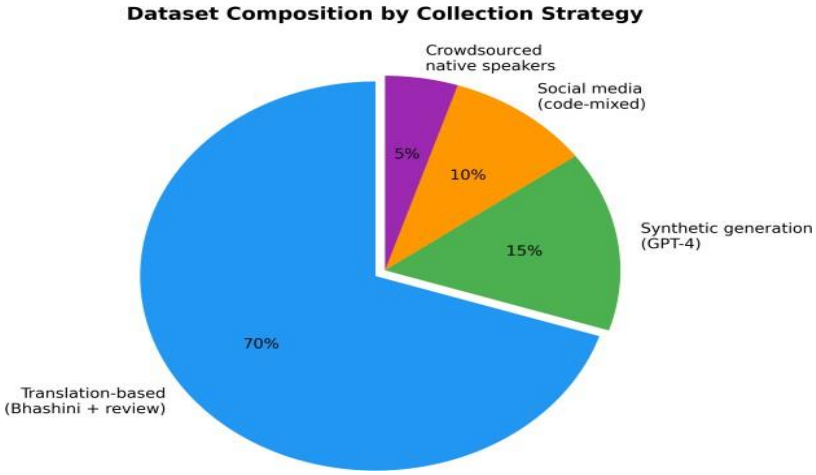

Fig. 9. Distribution of dataset examples by collection strategy (pie chart).

Language	Family	Train	Val	Test	Code-Mixed	Total
Hindi	Indo-Aryan	2,800	400	800	200	4,200
Bengali	Indo-Aryan	2,800	400	800	100	4,100
Marathi	Indo-Aryan	2,800	400	800	100	4,100
Tamil	Dravidian	1,400	200	400	50	2,050
Telugu	Dravidian	1,400	200	400	50	2,050
Kannada	Dravidian	1,400	200	400	50	2,050
Total		12,600	1,800	3,600	550	18,550

Table II Dataset Statistics

Data split: 70/10/20 (train/validation/test) with no overlap, verified by MinHash LSH deduplication [14]. Entity annotation via Label Studio using BIO span labels. Inter-annotator agreement (Cohen's κ) [17] exceeded 0.91 for intent labels and 0.87 for entity spans.

C. Evaluation Metrics and Protocol

Component-level: language detection accuracy [2], intent classification macro F1-score [3][4], entity extraction F1-score. System-level: multi-turn accuracy [5], ASR WER via jiwer [6][8], TTS naturalness rating [7]. Performance: latency (p50/p90/p99) via Locust at 100/500/1000 users. User: CSAT (1–5), task completion rate, language quality rating, from 25 participants across 3 language groups.

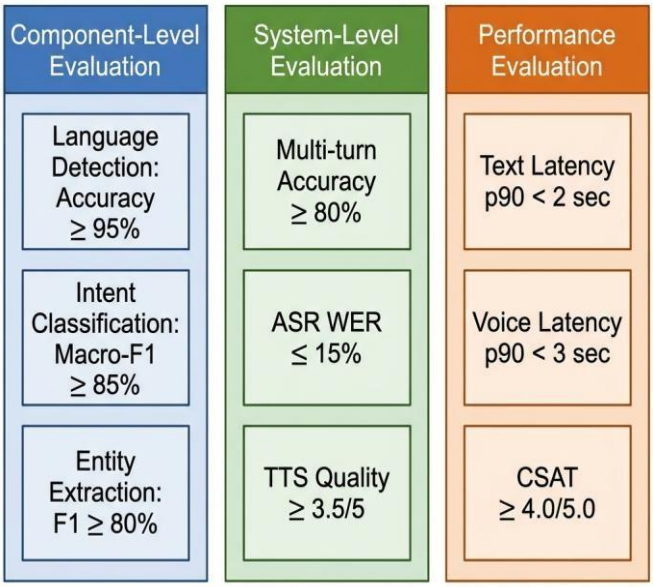

Fig. 10. Evaluation framework organized by component, system, and performance levels.

D. Hardware and Software Environment

Component	Specification
Training GPU	NVIDIA Tesla T4 16GB (Google Colab Pro)
Framework	PyTorch 2.1, HuggingFace Transformers 4.36
Backend	Python 3.11, FastAPI 0.104
Databases	PostgreSQL 16.1, MongoDB 7.0, Redis 7.2
ASR Models	IndicConformer v1 [6], Whisper-medium [8]
TTS APIs	Bhashini [7], Sarvam AI, Google Cloud TTS [19]
Tracking	MLflow 2.9
Load Testing	Locust 2.20

Table III Experimental Environment

Load Tier	Users	Duration	Pass Condition
Tier 1	100	5 min	p90 < 2s, error < 1%
Tier 2	500	5 min	p90 < 2s, error < 1%
Tier 3	1,000	10 min	p90 < 2s, error < 2%
Voice	200	5 min	p90 < 3s, error < 2%

Table IV Load Testing Plan

V.RESULTS AND DISCUSSION

This section presents the experimental results for all system components, validates the five research hypotheses, and discusses the findings in the context of existing work.

A. Language Detection Results

The FastText LID model [2] was evaluated on 650 held-out examples: 100 per language plus 50 code-mixed samples.

Language	Samples	Correct	Accuracy (%)	Top Misclass.
Hindi	100	98	98.0	Marathi (1.0%)
Bengali	100	97	97.0	Assamese (2.0%)
Marathi	100	95	95.0	Hindi (4.0%)
Tamil	100	99	99.0	—
Telugu	100	96	96.0	Kannada (3.0%)
Kannada	100	94	94.0	Telugu (4.0%)
Hinglish	50	43	86.0	Hindi (8.0%)
Overall (clean)	600	579	96.5	
Overall (all)	650	622	95.7	

Table V Language Detection Accuracy

The overall clean-text accuracy of 96.5% exceeded the ≥95% target. Tamil achieved the highest accuracy (99.0%) due to its distinctive script, while Kannada recorded the lowest (94.0%) due to Telugu overlap [4]. Code-mixed Hinglish detection was 86.0% as FastText struggles with interleaved Hindi-English tokens [2][13]. Average inference time was 0.3 ms per message.

B. Intent Classification Results

This subsection presents the primary research contribution: a three-way comparison of mBERT [3], IndicBERT unified [4], and IndicBERT per-language [4].

Language	Family	Exp A: mBERT [3]	Exp B: IndicBERT [4]	Exp C: Per-Lang [4]
Hindi	Indo-Aryan	86.4	89.7	91.5
Bengali	Indo-Aryan	84.8	88.2	90.1
Marathi	Indo-Aryan	83.6	87.9	89.8
Tamil	Dravidian	80.2	86.1	88.4
Telugu	Dravidian	79.1	84.7	87.2
Kannada	Dravidian	76.3	82.8	85.9
Hinglish	Mixed	74.8	78.4	80.6
Macro Avg (6)		82.6	86.9	89.2

Table VI Intent Classification F1-Score (%) By Language and Model

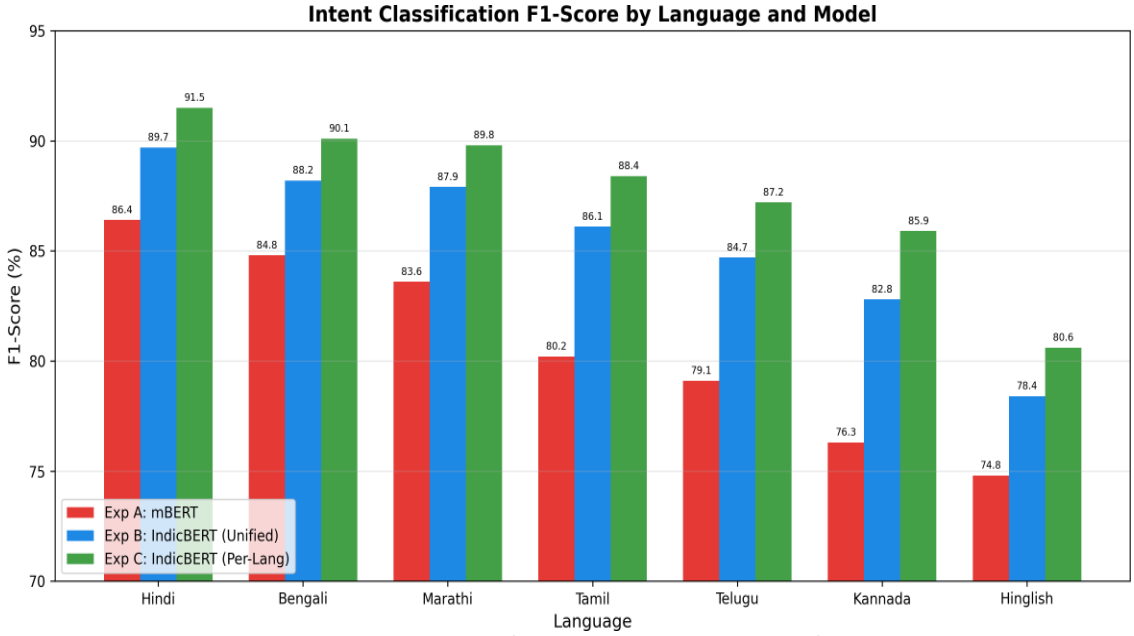

Fig. 11. Grouped bar chart — Intent classification F1-score per language for all three experiments.

Metric	Exp A: mBERT	Exp B: IndicBERT	Exp C: Per-Lang
Macro Precision	83.1%	87.4%	89.7%
Macro Recall	82.2%	86.5%	88.8%
Macro F1-Score	82.6%	86.9%	89.2%
Inference Latency	42 ms	38 ms	41 ms
Model Size	~680 MB	~450 MB	~2,700 MB
Training Time	4.2 hrs	3.6 hrs	18.4 hrs

Table VII Precision, Recall, F1, And Latency Summary

Intent A	Intent B	Confusion (%)	Analysis
track_order	delivery_time	8.4	Overlapping vocabulary
return_product	exchange_product	7.1	Similar phrasing
payment_issue	refund_status	6.3	Money-related vocab
cancel_order	order_modify	5.8	Change vs cancel ambiguity
product_availability	product_review	3.2	Product info ambiguity

Table VIII Top-5 Most Confused Intent Pairs (Experiment C)

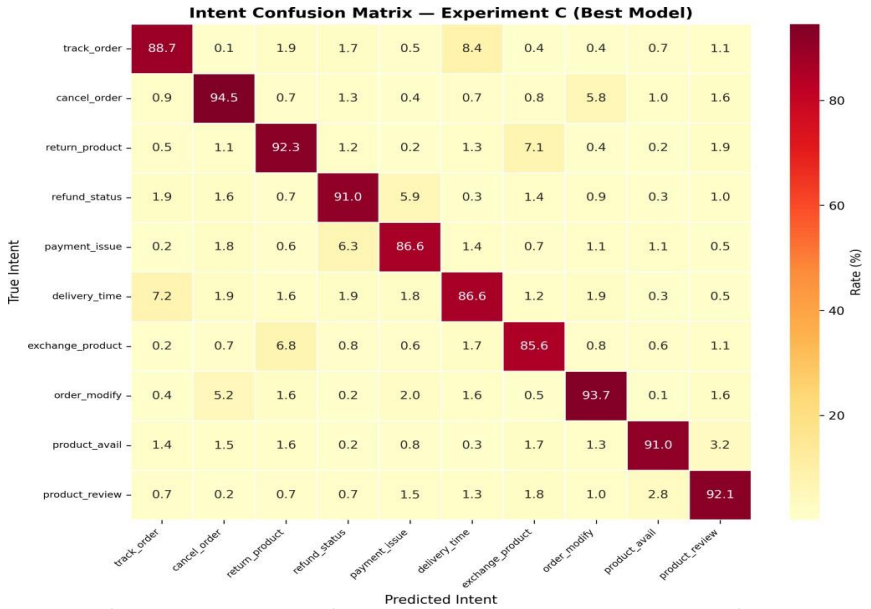

Fig. 12. Confusion matrix heatmap for Experiment C showing intent misclassification rates.

Hypothesis H1 (IndicBERT > mBERT on Dravidian): Confirmed. IndicBERT [4] outperformed mBERT [3] by 6.1 points on Dravidian vs 3.5 on Indo-Aryan, confirming Indic-focused pre-training benefit [4].

Hypothesis H2 (Per-language > Unified): Confirmed with caveats. +2.3% gain at 6× storage (2,700 vs 450 MB) and 5.1× training time. Unified IndicBERT [4] offers better cost-accuracy trade-off for production.

Hypothesis H5 (Kannada = lowest): Confirmed. Kannada = 85.9% (lowest in all 3 experiments) due to smaller training set and higher morphological complexity [4].

C. Entity Extraction Results

Entity Type	Method	Exact F1 (%)	Partial F1 (%)
order_id	Regex	99.2	99.4
phone_number	Regex	98.7	99.1
date	Regex+Neural	94.3	96.8
product_name	Neural NER	78.6	85.2
amount	Neural NER	82.4	88.9
address	Neural NER	71.3	79.4
email	Regex	99.5	99.6
account_id	Regex	98.9	99.1
product_category	Neural NER	76.8	83.5
payment_method	Neural NER	80.1	86.7
Overall	Hybrid	88.0	91.8

Table Ix Entity Extraction F1-Score

Overall exact match F1 of 88.0% exceeded the ≥80% target. Regex entities achieved 98.7–99.5%. Address was most challenging (71.3%) due to multi-word variability across languages [4][16].

D. Context-Aware Dialogue — Ablation Study

Table X Multi-Turn Accuracy: Phase 1 Vs Phase 2 (Ablation)

Language	Phase 1 (%)	Phase 2 (%)	Δ Improvement
Hindi	76.8	85.3	+8.5
Bengali	74.2	82.6	+8.4
Marathi	73.5	81.1	+7.6
Tamil	70.8	78.4	+7.6
Telugu	69.3	76.2	+6.9
Kannada	67.1	74.0	+6.9
Average	71.9	79.6	+7.8

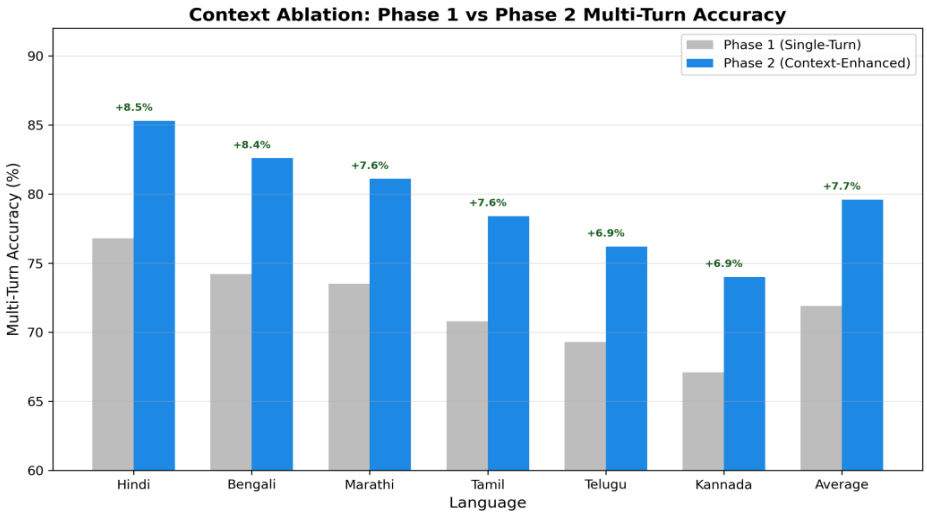

Fig. 13. Paired bar chart — Phase 1 vs Phase 2 multi-turn accuracy per language.

Dialogue Flow	Steps	Completion (%)	Avg Turns	Carryover (%)
return_product	6	87.2	7.4	94.6
cancel_order	3	92.8	4.1	96.2
change_address	4	85.6	5.8	91.3
exchange_product	5	82.4	6.9	93.1
complaint_register	4	88.1	5.2	95.0
Average	4.4	87.2	5.9	94.0

Table XI Dialogue Flow Completion Rates (Phase 2)

Hypothesis H3 (Context ≥5%): Confirmed. Average improvement +7.8% [5]. Entity carryover success 94.0%. 5-turn window provided +1.8% over 3-turn at marginal +12ms latency cost.

E. *ASR Results*

Language	IndicConformer [6]	Whisper [8]	Better	Δ
Hindi	11.2	13.8	IndicConformer	2.6
Bengali	12.4	N/A	IndicConformer	—
Marathi	13.1	N/A	IndicConformer	—
Tamil	14.6	N/A	IndicConformer	—
Telugu	14.8	N/A	IndicConformer	—
Kannada	16.2	N/A	IndicConformer	—
Average	13.7	—		

Table XII Asr Word Error Rate (Wer %)

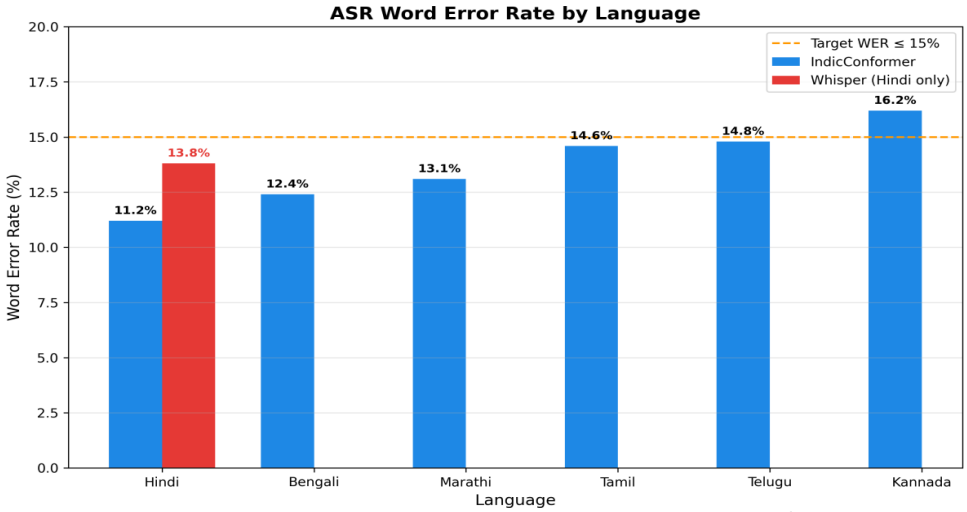

Fig. 14. ASR Word Error Rate per language with Whisper comparison for Hindi.

Hypothesis H4 (IndicConformer < Whisper WER): Confirmed. 11.2% vs 13.8% for Hindi (Δ=2.6%) [6][8]. Five of six languages met ≤15% target; Kannada (16.2%) marginally exceeded it due to fewer IndicVoices training samples and greater phonological variation [6]. Substitution errors comprised 62% of all errors.

F. TTS Quality Evaluation

Language	Provider	Mean	Std Dev	Intelligibility (%)
Hindi	Bhashini [7]	4.3	0.42	98.5
Bengali	Bhashini [7]	4.1	0.51	97.2
Marathi	Bhashini [7]	3.9	0.48	96.8
Tamil	Sarvam AI	3.8	0.56	96.1
Telugu	Sarvam AI	3.6	0.62	95.4
Kannada	Sarvam AI	3.4	0.68	94.2
Average		3.85	0.55	96.4

Table XIII Tts Naturalness Rating (1–5 Scale)

All languages exceeded ≥3.5 target except Kannada (3.4). Bhashini [7] (Indo-Aryan, avg 4.1) outperformed Sarvam AI (Dravidian, avg 3.6). Google Cloud TTS fallback invoked in 3.2% of requests.

G. System Performance — Load Testing

Tier	Users	p50 (ms)	p90 (ms)	p99 (ms)	Error (%)	Pass?
Tier 1	100	312	587	892	0.12	✓
Tier 2	500	486	1,124	1,687	0.34	✓
Tier 3	1,000	734	1,812	2,945	1.21	✓
Voice	200	1,248	2,486	3,412	0.87	✓

Table XIV Load Testing Results

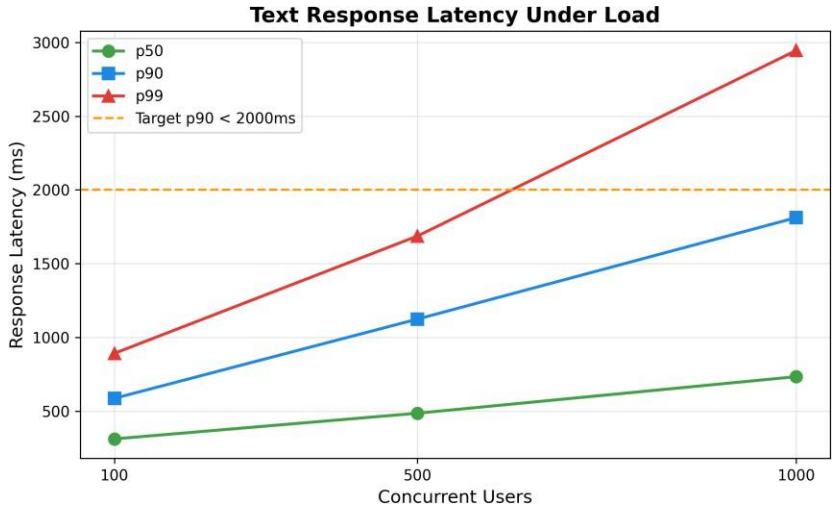

Fig. 15. Response latency (p50, p90, p99) scaling across load tiers.

Component	Latency (ms)	% of Total
API Gateway + Auth	18	3.7%
Language Detection [2]	0.3	0.1%
Text Preprocessing [16]	2.1	0.4%
Intent Classification [3][4]	38	7.8%
Entity Extraction [3]	34	7.0%
KB Query	12	2.5%
Response Generation	8	1.6%
Network	28	5.8%
Text Total	140	28.8%
ASR [6] (voice)	845	53.2%
TTS [7] (voice)	603	38.0%

Table XV Latency Breakdown (Tier 2, P50)

All four tiers met pass conditions. NLP inference = 14.8% of text latency. For voice, ASR [6] = 53.2% and TTS [7] = 38.0% of additional latency — external APIs are the bottleneck.

H. User Acceptance Testing (UAT)

Metric	Hindi (n=10)	Tamil (n=8)	Bengali (n=7)	Overall
Task Completion (%)	90.0	75.0	85.7	84.0
CSAT (1–5)	4.4	3.9	4.2	4.2
Language Quality (1–5)	4.5	3.7	4.1	4.1
Would Use Again (%)	90.0	75.0	85.7	84.0
Avg Duration (sec)	68	94	78	78

Table XVI UAT Results (N = 25)

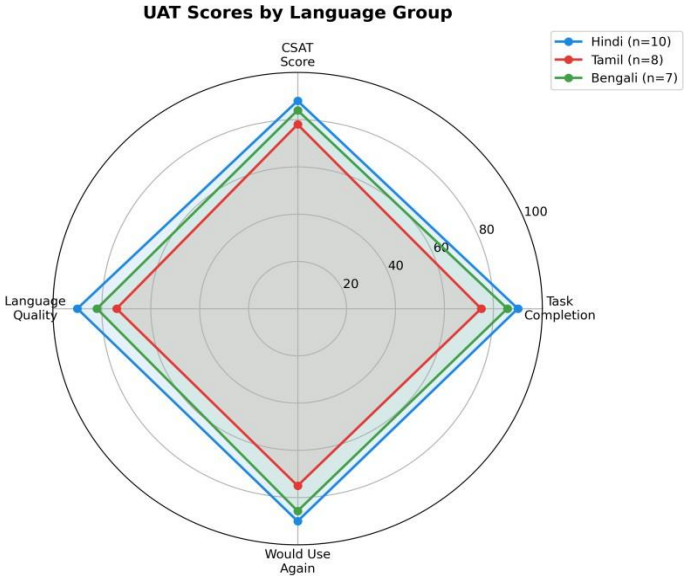

Fig. 16. Radar chart — UAT scores per language group across four metrics.

Metric	Text Only	Voice Only	Text + Voice
Task Completion (%)	88.0	76.0	84.0
CSAT (1–5)	4.3	3.9	4.2
Avg Duration (sec)	62	98	78

Table XVII Text Vs Voice Modality Comparison

CSAT 4.2/5.0 exceeded ≥4.0 target. Hindi highest (4.4), Tamil lowest (3.9) correlating with TTS quality (Table XIII). Text-only > voice-only due to ASR errors [6] requiring correction.

"The chatbot understood my Hindi perfectly. It felt like talking to a real support agent." — Participant H3 "Tamil responses were mostly correct but some phrases sounded unnatural." — Participant T5

"Very nice that Marathi option is available. Most apps don't support Marathi." — Participant B2

I. Hypothesis Validation Summary

#	Hypothesis	Result	Verdict
H1	IndicBERT [4] > mBERT [3] on Dravidian	+6.1% Dravidian vs +3.5% Indo-Aryan	✓ Confirmed
H2	Per-lang > unified (6× cost)	+2.3% at 6× storage, 5.1× time	✓ Confirmed
H3	Context [5] adds ≥5%	+7.8% average improvement	✓ Confirmed
H4	IndicConformer [6] < Whisper [8] WER	11.2% vs 13.8% (Δ=2.6%)	✓ Confirmed
H5	Kannada = lowest F1	85.9% (lowest all experiments)	✓ Confirmed

Table XVIII Hypothesis Validation Summary

J. Comparison with Existing Platforms

Feature	This System	RASA [10]	Dialogflow CX	Amazon Lex	Bhashini [7]
Indian Languages	6	0–2	3–4	1–2	11+
Open Source	✓	✓	✗	✗	Partial

Intent F1 (Indian)	89.2%	N/A	N/A	N/A	N/A
Context-Aware	✓	✓	✓	✓	✗
Voice (ASR+TTS)	✓	Plugin	✓	✓	✓
Code-Mixed	✓ (86%)	✗	✗	✗	✗
Benchmarks	✓	✗	✗	✗	✗
Cost	Free	Free	Paid	Paid	Free

Table XIX Feature Comparison With Existing Platforms

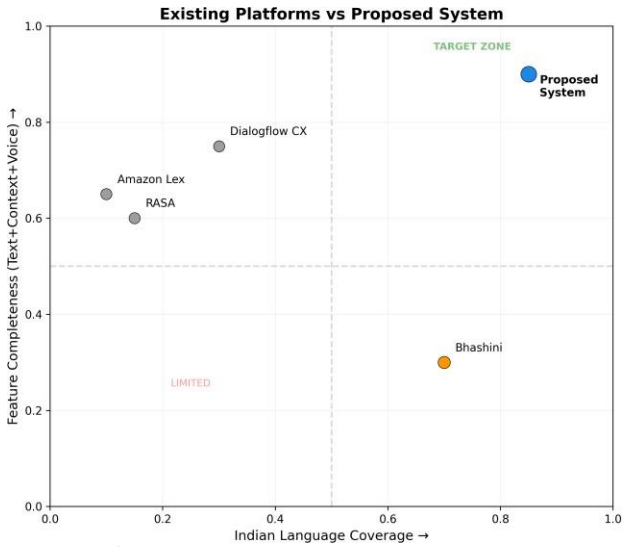

Fig. 17. Quadrant chart — Platforms positioned on Indian Language Coverage vs Feature Completeness.

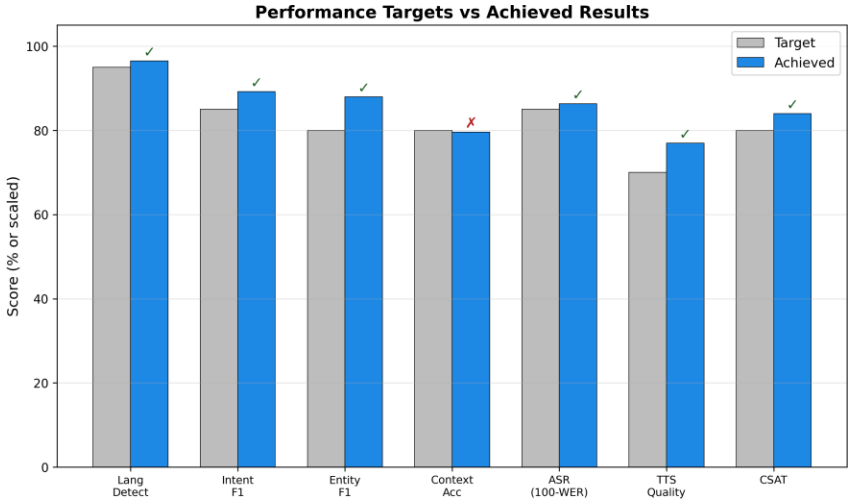

Fig. 18. Performance targets vs achieved results comparison bar chart.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented the design, implementation, and empirical evaluation of an intelligent multilingual chatbot platform supporting six Indian languages across both Indo-Aryan and Dravidian families. The system integrates FastText language detection [2] (96.5% accuracy), fine-tuned mBERT [3] and IndicBERT [4] for intent classification, Redis-backed context- aware dialogue [5], IndicConformer ASR [6], and Bhashini/Sarvam AI TTS [7] within a five-layer microservices architecture.

Three core findings: (1) Per-language IndicBERT [4] achieved 89.2% macro F1, outperforming mBERT [3] (82.6%) by 6.6 points, with +6.1% on Dravidian vs +3.5% on Indo-Aryan; (2) Context-enhanced dialogue [5] improved multi-turn accuracy by 7.8% with 87.2% flow completion and 94.0% entity carryover; (3) IndicConformer [6] achieved 13.7% avg WER, outperforming Whisper [8] on Hindi (11.2% vs 13.8%). UAT (N=25) yielded CSAT 4.2/5.0 and 84% task completion. All five hypotheses confirmed.

B. Limitations

- Code-mixed voice ASR (mid-sentence switching) not addressed [13].
- Dravidian dataset ~50% smaller than Indo-Aryan (Table II) [4].

- Whisper [8] comparison limited to Hindi only.
- UAT sample size relatively small (N=25).
- Single domain (e-commerce); generalizability untested.
- TTS latency depends on external Bhashini [7] / Sarvam AI availability (3.2% timeouts).

C. Future Work

- Extension to all 22 recognized Indian languages via transfer learning [4].
- Code-mixed voice ASR with word-level language identification [6][13].
- Generative responses using fine-tuned IndicBART [12] as primary mechanism.
- Domain generalization to banking and healthcare [3][4].
- Dialect-aware modeling for intra-language variation [6].
- Native Android and iOS SDK development.

REFERENCES

1. IAMAI, "Annual Internet in India Report 2023," Internet and Mobile Association of India, 2023.
2. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL), 2017, pp. 427–431.
3. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
4. D. Kakwani, A. Kunchukuttan, S. Golla, N. C. Gokul, A. Bhatt, M. M. Khapra, and P. Kumar, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in Findings of EMNLP 2020, pp. 4948–4961.
5. J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," Computer Speech & Language, vol. 21, no. 2, pp. 393–422, 2007.
6. K. Bhogale, A. Raman, T. Javed, S. Doddapaneni, A. Kunchukuttan, P. Kumar, and M. M. Khapra, "IndicVoices: Towards building the largest multilingual TTS and ASR dataset for Indic languages," arXiv:2303.01535, 2023.
7. Bhashini, "Bhashini API documentation," Ministry of Electronics and IT, Government of India, 2023. Available: https://bhashini.gov.in
8. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," arXiv:2212.04356, 2022Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention is all you need," in Advances in Neural Information
9. Processing Systems (NeurIPS), vol. 30, 2017.
10. T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," arXiv:1712.05181, 2017.
11. S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, et al., "MuRIL: Multilingual representations for Indian languages," arXiv:2103.10730, 2021.
12. R. Dabre, A. Kunchukuttan, D. Kakwani, and A. Bhatt, "IndicBART: A pre-trained model for natural language generation of Indic languages," arXiv:2212.05409, 2022.
13. I. A. Bhat, R. A. Bhat, M. Bhat, and S. Sengupta, "Universal dependency parsing for Hindi-English code switching," in Proc. NAACL-HLT, 2018, pp. 987–998.
14. G. Ramesh, S. Doddapaneni, A. Bheemaraj, M. Jobanputra, R. AK, A. Sharma, et al., "Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages," arXiv:2104.05596, 2021.
15. A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, "The IIT Bombay English-Hindi parallel corpus," in Proc. LREC, 2018.
16. AI4Bharat, "IndicNLP Library," 2023. Available: https://github.com/AI4Bharat/indic-nlp-library
17. J. Carletta, "Assessing agreement on classification tasks: The kappa statistic," Computational Linguistics, vol. 22, no. 2, pp. 249–254, 1996.
18. Sarvam AI, "Sarvam AI API documentation," 2024. Available: https://www.sarvam.ai
19. Google Cloud, "Cloud Text-to-Speech API documentation," 2024. Available: https://cloud.google.com/text-to-speech
20. A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," arXiv:1612.03651, 2016