Multilingual Conversational AI for Financial Assistance: Bridging Language Barriers in Indian FinTech

Anonymous ACL submission

Abstract

India's linguistic diversity presents both opportunities and challenges for fintech platforms. While the country has 31 major languages and over 100 minor ones, only 10% of the population understands English, creating barriers to financial inclusion. We present a multilingual conversational Al system for a financial assistance use case that supports code-mixed languages like Hinglish, enabling natural interactions for India's diverse user base. Our system employs a multi-agent architecture with language classification, function management, and multilingual response generation. Through comparative analysis of multiple language models and real-world deployment, we demonstrate significant improvements in user engagement while maintaining low latency overhead (4-8%). This work contributes to bridging the language gap in digital financial services for emerging markets.

1 Introduction

The digital landscape of India is undergoing a transformation of unprecedented scale, characterized by rapid growth and profound linguistic diversity. This dual nature presents both immense opportunities and significant challenges for technology platforms, particularly in critical sectors like finance. Recent data indicates that India's internet user base has surged to 886 million, with an 8% year-over-year growth predominantly driven by users in rural areas. Projections suggest this figure could surpass 900 million by 2025, cementing India's position as one of the world's largest and most dynamic digital markets. (IAMAI, 2024)

However, the most defining characteristic of this market is its linguistic fabric. A staggering 90% of the population does not possess proficiency in English, the traditional lingua franca of the digital world(IAMAI, 2024). This reality is reflected in user behavior: nearly all Indian internet users access content in one of the nation's 22 official languages and hundreds of dialects, and over half of all urban users express a preference for consuming content in their native languages. Furthermore, historical data reveals that 90% of new internet adopters are non-English speakers, underscoring a clear and irreversible trajectory: the "next wave of online content will be linguistically diverse".

This linguistic imperative is particularly acute in the financial technology (fintech) sector. India's asset management industry has witnessed remarkable expansion, with significant contributions coming from beyond the traditional metropolitan hubs. Over the past six years, Tier 2 and Tier 3 cities have increased their mutual fund Assets Under Management (AUM) by 13% (NSE), indicating a growing appetite for investment products among a new class of retail investors. This growth, however, is severely constrained by a persistent language barrier. While the majority of digital banking and fintech services in India are offered exclusively in English or, at best, Hindi, this overlooks the linguistic realities even in major financial hubs; for instance, top AUM states like Maharashtra, New Delhi, and Karnataka, despite their robust and diversified investment portfolios, are home to large populations primarily speaking languages such as Marathi, Kannada, and various regional dialects. (AMFI, 2025)

We are building technology to democratize access to quality investment advice for retail investors, combining AI with quantitative modules. Our conversational AI engages with users naturally, allowing them to ask questions and better understand recommendations—a critical feature in a market where financial literacy remains a barrier. Supporting multiple languages ensures that we can bridge the gap where traditional distributors cannot, opening access to a large and fast-growing segment of India's retail investment market.

The key contributions of this work are threefold:

A Novel Multi-Agent Architecture: We propose and implement a multi-agent framework that effectively orchestrates language classification, domain-specific function management, and multilingual response generation for complex financial dialogues. This architecture provides a robust and scalable solution for handling the multi-faceted nature of financial conversations.

Empirical Model Analysis for a Niche Domain: We provide a comparative analysis of various large and small language models for the specific task of Hinglish financial conversation. Our findings demonstrate the superiority of domain-adapted models like Indic-BERT for specialized tasks such as language detection over general-purpose models, offering valuable insights for practitioners building similar systems.

Real-World Deployment Insights: We demonstrate the system's practical viability through a proof-of-concept deployment. By analyzing user interactions and engagement metrics, we report significant improvements in user engagement and provide a qualitative analysis of how users interact with a code-mixing financial chatbot, validating its effectiveness in a real-world setting.

2 Related Work

Our work builds on four key research areas: multilingual natural language processing (NLP) for Indian languages, the study of code-switching in conversational AI and the application of AI in the financial domain.

2.1 Advances in Multilingual NLP for Indian Languages

The rapid growth of India's digital ecosystem has spurred significant research into NLP for Indic languages. A primary focus has been the development of large-scale, pre-trained multilingual language models capable of understanding the nuances of the region's diverse linguistic landscape. Foundational models such as mBERT, XLM-RoBERTa (Conneau et al., 2020), and more specialized models like IndicBERT (Kakwani et al., 2020), MuRIL (Multilingual Representations for Indian Languages) (Khanuja et al., 2021) and IndicTransformers (Jain et al., 2020) have been instrumental. These models are typically pre-trained on large corpora spanning multiple Indian languages, enabling effective transfer learning for various downstream tasks, including text classification, named entity recognition, and question answering.

IndicBERT, for instance, was pre-trained on a corpus of 11 major Indian languages from the Indo-Aryan and Dravidian families, making it particularly well-suited for tasks requiring cross-lingual understanding within the Indian context. Similarly, MuRIL was trained on 17 Indian languages alongside English, leveraging parallel and transliterated corpora to enhance its performance. These models address a critical challenge in Indic NLP: the relative scarcity of monolingual data for many Indian languages compared to high-resource languages like English.

A pivotal study by Dhamecha et al. (2021) (Dhamecha et al., 2021) from IBM Research explored the role of language relatedness in multilingual fine-tuning. Their work demonstrated that fine-tuning a model on a carefully selected subset of related languages (in their case, from the Indo-Aryan family) can yield significantly better performance than fine-tuning on individual languages or on a larger, more diverse set of languages. This finding suggests that linguistic proximity enables positive knowledge transfer, a principle that can guide the strategic expansion of multilingual systems.

2.2 Code-Switching and Code-Mixing in Conversational AI

Code-switching (CS) or code-mixing (CM), the practice of alternating between two or more languages within a single conversation or utterance, is a pervasive linguistic phenomenon in multilingual communities. For conversational AI systems to feel natural and engaging to a large segment of the Indian population, the ability to understand and reciprocate code-mixing is not a luxury but a necessity.

Pioneering user studies in this area have provided empirical justification for this claim. A mixed-method study by (Bawa et al., 2020) from Microsoft Research conclusively found that "multilingual users strongly prefer chatbots that can code-mix". Their experiment compared monolingual bots with bots employing different code-mixing strategies and found that user ratings for naturalness and conversational ability were significantly higher for code-mixing bots. A key finding was the effectiveness of a "Nudge" policy, where the bot subtly introduces code-mixed cues and adapts based on the user's reciprocation.

Despite its importance, handling code-mixed text remains an active and challenging area of research. Many state-of-the-art LLMs, while powerful in monolingual contexts, are not yet adept code-switchers and can struggle with the syntactic and semantic complexities of mixed-language input. To address this, recent research has explored advanced fine-tuning techniques. A notable approach is the CHAI framework (Zhang et al., 2025), which proposes using reinforcement learning from AI feedback (RLAIF) to improve an LLM's capability to handle code-mixed tasks like machine translation.

2.3 Conversational AI in the Financial Domain

The financial services industry has been an early and enthusiastic adopter of AI, deploying it for a wide range of applications including algorithmic trading, risk management, fraud detection, and automated customer service. Conversational AI, in the form of chatbots and voice bots, has become a common feature, aimed at improving operational efficiency, reducing costs, and providing 24/7 customer availability.

In the Indian context, several leading banks and fintech companies have deployed multilingual chatbots. Notable examples include the State Bank of India's SIA, HDFC Bank's EVA, and ICICI Bank's iPal. These systems are designed to handle routine banking queries in multiple Indian languages (Kediya et al., 2023; Bansal et al., 2024; Kakwani et al., 2025; Kanchan et al.; Sachdeva and Dhingra, 2024; Ray and Anirudhan, 2023; Saleem and Mathew). A recent trend is the collaboration with government-led language technology platforms; for instance, Federal Bank partnered with Bhashini to enable its chatbot, Feddy, to support 14 languages, aligning with the national push for digital financial inclusion through vernacular support.

However, a review of both industry deployments and academic literature reveals a gap. While many systems are described as "multilingual", this often refers to the ability to conduct a conversation in one of several supported monolingual modes. There is significantly less documented work on systems that can handle dynamic, intra-sentential code-mixing for the specific, high-stakes domain of financial advisory.

3 System Description

Our conversational AI system is engineered to serve as a financial guidance system for users in the diverse Indian linguistic landscape. The architecture is a modular, multi-stage pipeline designed to decouple linguistic complexity from core financial logic, ensuring robustness, scalability, and maintainability. The system now processes a user's query through four primary stages: Language Classification, Orchestration, Tool Execution, and Response Generation as described in Figure 1

3.1 Language Classifier

The entry point to our system is a dedicated Language Classification module. Its function is to perform a rapid and accurate analysis of the user's input to identify the primary language (e.g., English, Hindi, Marathi, Gujarati) and to detect the presence of code-mixing (e.g., "Hinglish"). The key requirements for this component are extremely low latency and high accuracy, as its output dictates the behavior of all downstream modules.

3.2 Orchestrator

The core intelligence of our system resides in the Orchestrator, a Large Language Model (LLM) engineered to perform two critical tasks:

Query Rephrasing & Normalization: The Orchestrator first normalizes the user's raw input into a standardized, machine-readable English format. This step is pivotal for handling code-mixed queries by creating a language-agnostic representation.

Intent Classification: The Orchestrator then performs intent classification on the normalized query to select the appropriate financial tool required to fulfill the user's request. This ensures the core logic operates on a consistent data structure.

3.3 Specialized Financial Tools

Our system utilizes a suite of specialized worker agents or "tools" to execute financial tasks. These tools are heterogeneous in nature:

- Software Modules: Deterministic functions that execute specific, programmatic tasks such as retrieving data from a portfolio database or calling a stock price API.

- LLM-Powered Agents: A combination of LLMs and code for more dynamic use cases

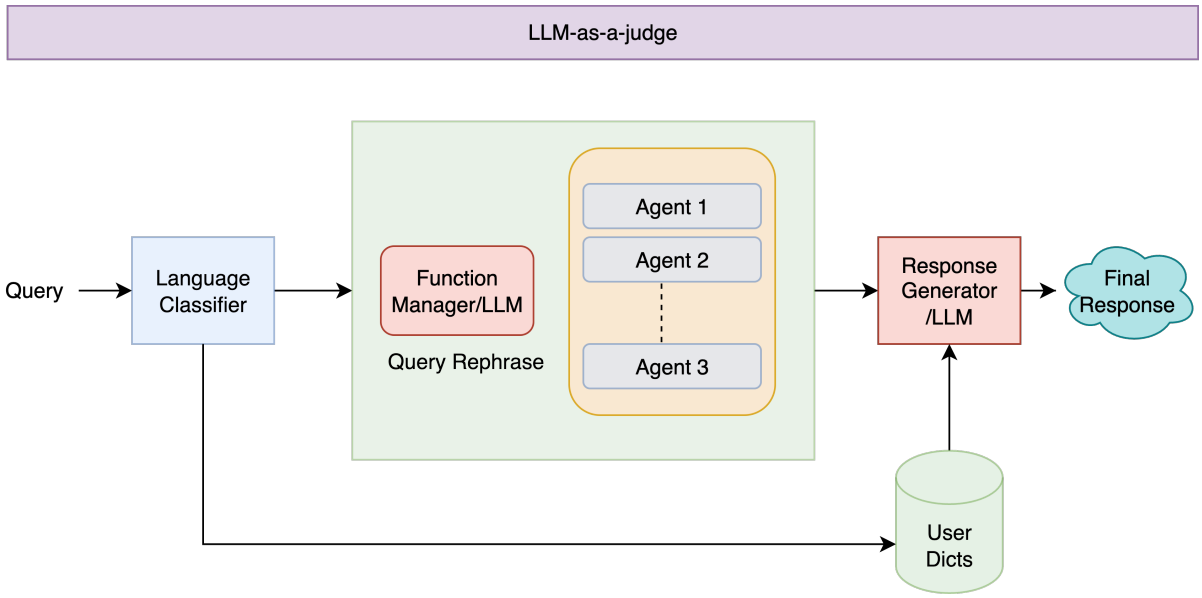

Figure 1: System architecture showing the flow from user query through language classification, function management, agent selection, and response generation for supporting multilingual queries

that require nuanced understanding or complex data synthesis, such as advanced fund comparison or generating qualitative security evaluations.

Current tools handle a range of functionalities including portfolio analytics, securities search, fund screening, and answering general financial queries.

3.4 Response Generation Module

The final module constructs the reply presented to the user. It receives two key inputs: (1) the structured data output from the executed financial tool(s) (in English), and (2) the original language tag (e.g., lang='hindi') from the initial classifier. This module employs a multilingual LLM that synthesizes these inputs to generate a coherent, context-aware, and natural-sounding response in the user's original language.

4 Approach to Multilingual and Code-Mixed Dialogues

Our initial goal was to extend our existing English-only financial advisory platform to support India's multilingual user base. We detail the iterative, empirically-driven approach we took to achieve this.

4.1 Baseline Performance and Problem Analysis

We first evaluated the baseline performance by subjecting our existing system to multilingual (Hindi, Marathi, Gujarati) and code-mixed queries in a zero-shot setting. The results were poor, with a 20-45% drop in the end-to-end task success rate compared to pure English queries. An analysis revealed that errors were systemic and cascaded through the workflow: the orchestrator failed to comprehend the intent, the tool-use modules received incorrect inputs, and the Response Generator produced irrelevant output. This demonstrated that simply using a powerful base LLM was insufficient.

Table 1: Initial set of multilingual LLMs evaluated for Indic language support

Model	Parameters	Architecture	Hindi Support
Llama 3.1	8B, 70B, 405B	Llama 3.1	Limited
Hermes 3	8B, 70B, 405B	Llama 3.1	Yes
Aya Expanse	8B, 32B	-	Yes
Airavata	7B	Llama 2	Yes
sarvam-2b-v0.5	2B	-	Yes
LLama3-Gaja-Hindi	8B	Llama 3	Yes

4.2 Experiments

4.2.1 Dedicated Classification and Prompting

Our first attempt to remedy this involved two architectural changes:

- Introducing a Language Classifier: To effectively handle multilingual inputs from the start, we introduced a lightweight classifier at the beginning of the pipeline. We conducted a detailed evaluation of several models to identify the optimal classifier that could manage pure and code-mixed languages with minimal latency. Indic-BERT (Kakwani et al., 2020), a model pre-trained on 11 Indian languages (Kakwani et al., 2020), demonstrated substantially higher accuracy and F1-scores on complex code-mixed text, with a latency under 20ms. Qwen2.5-0.5B (Xu et al., 2025; Bai et al., 2025) model was the second best lightweight model [A detailed comparison of classifier models is presented in Table 2].

- Language-Specific Prompt Templates: We created curated prompt templates for the Orchestrator for each language we intended to support.

Table 2: Language Detection Performance Comparison

Query Type	Model	Accuracy (%)	F1-Score
Pure English	Qwen2.5-0.5B	99.5	0.99
	Indic-BERT	99.8	1.00
Pure Hindi	Qwen2.5-0.5B	98.2	0.98
	Indic-BERT	99.5	0.99
Hinglish (General)	Qwen2.5-0.5B	85.4	0.84
	Indic-BERT	97.1	0.97
Hinglish (Financial)	Qwen2.5-0.5B	63.7	0.61
	Indic-BERT	95.8	0.96

While this approach improved performance on pure language queries, it consistently failed on more nuanced code-mixed inputs. For example, a query like ”mera equity exposure kitna hai?” would be correctly classified as Hinglish, but the Orchestrator, despite the Hinglish-specific prompt, would fail to reliably associate the English term ”equity exposure” within a Hindi sentence structure to the `get_portfolio_analytics` tool. This revealed that a deeper semantic normalization was required.

4.2.2 Decoupling Language from Logic via Query Rephrasing

The critical insight from the failure of our first iteration was the realization that the entire system does not need to be multilingual, only the user-facing layers do. The core financial logic within the tools could, and should, remain language-agnostic for simplicity and reliability.

To achieve this, we implemented the query rephrasing and normalization step within the Orchestrator, as described in Section 3.2. This step acts as a translation layer, effectively creating an abstraction between the user's linguistic expression and the system's logical operations. By converting all inputs into a canonical English representation before tool selection, we decoupled the robust, pre-existing financial tools from the complexities of multilingual understanding.

4.3 Final System Evaluation

We validated this final architecture against a ”golden” test set of multi-turn conversations covering various intents across all supported languages and code-mixing patterns. Task success was measured using a combination of deterministic and non-deterministic metrics:

- Intent Tool Call Accuracy: An exact-match assertion to verify that the correct intent and tool parameters were derived.

- Response Quality: An LLM-as-a-judge framework to score the final generated response for correctness, coherence, and relevance against a reference answer.

The results confirmed that the final architecture, incorporating the Classifier -> Rephraser -> Dispatcher flow, successfully overcame the challenges of the baseline system, achieving task success rates on par with pure English queries across all tested languages.

4.4 Evaluating and Selecting the Response Generation Model

The Response Generation module requires a model that can generate high-quality, fluent responses in Indic languages while strictly adhering to the structured financial data it receives. We evaluated several state-of-the-art multilingual LLMs

(Teknium et al., 2024; Dang et al., 2024; Grattafiori et al., 2024; Sarvam, 2024) to find the best balance between conversational ability and instruction-following. Table 3 presents our findings.

Based on this analysis, Hermes-3-8B was selected as the core model for the Response Generator module. Its ability to follow complex instructions ensures financial accuracy, while its strong generative capabilities provide the natural conversational experience required by our users.

4.5 End-to-End System Evaluation

As introduced in Section 4.3, we validated the final architecture against a "golden" test set of multi-turn conversations covering various intents across all supported languages and code-mixing patterns. Task success was measured using a combination of deterministic and non-deterministic metrics:

- Intent & Tool Call Accuracy: An exact-match assertion to verify that the correct intent and tool parameters were derived by the Orchestrator.
- Response Quality: An LLM-as-a-judge framework to score the final generated response for correctness, coherence, and relevance against a reference answer.

For response quality, we took inspiration from G-Eval (Liu et al., 2023) for its lightweight setup and ease of adapting to our existing pipeline. To design our own rubric, we explored DeepEval's (Confident-AI, 2023) various metrics and strategies. This led us to define our own domain specific evaluation criteria viz

- Response completeness (1-5)
- Factual Accuracy (1-5)
- Consistent(to Query) Language Usage (True/False)
- Contextual Awareness (1-5)
- Scope Compliance (1-5)

The results confirmed that the final architecture, incorporating the Classifier -> Rephraser -> Dispatcher flow, successfully overcame the challenges of the baseline system, achieving task success rates on multilingual queries that were on par with pure English queries.

Furthermore, data from a proof-of-concept deployment with over 500 beta users demonstrated that bridging the language barrier directly translates to superior user outcomes and engagement. This is evident in Table 4, which presents results from human evaluation of response quality and Table 5, which details improvements in user engagement metrics

Error Analysis of Failure Cases

To understand the system's remaining weaknesses and guide future work, we manually reviewed and categorized 100 instances of failed or low-quality conversations from our deployment. The primary categories of errors are summarized in Table 6.

This analysis reveals that while our pragmatic, pipeline-based approach is highly effective, the system's robustness can decrease with increasing query complexity and linguistic ambiguity. The insights gained are invaluable for guiding future development, particularly in enhancing the orchestrator's multi-intent reasoning capabilities and improving the grounding mechanisms of the Response Generator to ensure factual faithfulness.

5 Conclusion and Future Directions

This paper presented a multilingual conversational AI system for financial guidance services in India using a novel multi-agent architecture that orchestrates language classification, intent recognition, and context-aware response generation for code-mixed financial dialogues. Our empirical analysis established the superiority of domain-adapted models like Indic-BERT for language detection and identified Hermes-3-8B as optimal for balancing instruction-following and multilingual response generation. Proof-of-concept deployment demonstrated significant real-world impact: 41% increase in task completion rates, 86% increase in average session length, and more than doubled user retention compared to English-only baselines. Future research will extend capabilities to other Indic languages using transfer learning principles and develop dialect-aware personalization models. This work provides a practical blueprint for building linguistically inclusive AI systems that can advance financial literacy and inclusion as India's vernacular-led digital economy continues to grow.

Table 3: Comparative analysis of multilingual LLMs for financial assistance tasks, evaluating response generation quality and instruction-following capabilities.

Model	Response Generation	Instruction Following
Sarvam-1.0-2B	Excellent quality in Indic languages	Poor capabilities, unreliable for structured financial tasks
Llama-3.1-8B	Good responses in Hindi	Struggles with instruction following, requires extensive prompt engineering
Aya-Expanse (8B)	Poor performance	Poor performance
Aya-Expanse (32B)	Better performance	Better performance, but impractical due to computational cost and latency
Hermes-3-8B	High-quality, fluent responses in Hindi and Hinglish	Advanced instruction-following abilities, optimal choice

Table 4: Human Evaluation of Response Quality (Mean Scores, 1-5 Scale)

Evaluation Criterion	TTR	Proposed system	Improvement
Fluency	3.2	4.5	+40.6%
Coherence	3.8	4.6	+21.1%
Helpfulness	4.1	4.7	+14.6%

Table 5: User Engagement Metrics (A/B Test)

Metric	English-Only	Multilingual	Improvement
Task Completion Rate	58%	82%	+41.4%
Avg. Session Length	4.2 turns	7.8 turns	+85.7%
30-Day Retention Rate	12%	25%	+108.3%

Ethical Considerations

The deployment of an AI system for financial assistance in a linguistically diverse country like India carries profound ethical responsibilities, particularly regarding bias, accountability, and data privacy. LLMs trained on internet data often absorb and amplify existing societal biases. In our context, this manifests as linguistic bias, where there's a risk of better performance for "standard" urban Hindi dialects compared to regional variations. It also leads to socio-economic bias, as training data skewed towards affluent customers may result in inappropriate advice for lower-income users. To address these concerns, we actively work to diversify training datasets, conduct regular bias audits, and maintain human-in-the-loop oversight for critical recommendations.

Limitations

Our current system focuses primarily on Hindi-English code-mixing and may not generalize to other Indian language combinations without significant adaptation. The evaluation is limited to financial assistance use-cases. Additionally, the system relies on existing multilingual models that may carry inherent biases affecting advice quality across varying user populations. The scarcity of high-quality code-mixed financial dialogue data remains a significant constraint for further model improvements.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude (Anthropic) and ChatGPT in order to: perform grammar and spelling checks, improve writing style and paraphrase and reword sections for clarity and conciseness. After using these tool(s)/service(s), the author(s) thoroughly reviewed, critically evaluated and edited all content to ensure accuracy and alignment with research objectives. The author(s) take(s) full responsibility for the publication's content.

References

AMFI. 2025. State-wise average assets under management in india. https://www.amfiindia.com/geographical-spreads. Accessed: 2025-05-04.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Chandni Bansal, Krishan Kumar Pandey, Rajni Goel, and Anuj Sharma. 2024. Analysis of barriers to ai banking chatbot adoption in india: An ism and micmac approach. *Issues in Information Systems*, 25(4):417–441.

Anshul Bawa, Pranav Khadpe, Pratik Joshi, Kalika Bali, and Monojit Choudhury. 2020. Do multilingual users prefer chat-bots that code-mix? let's nudge and find out! *Proceedings of the ACM on Human-Computer Interaction*, 41(CSCW1):1–23.

Confident-AI. 2023. Deepeval: The llm evaluation framework. https://github.com/confident-ai/deepeval. Accessed: 2025-07-04.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy,

Table 6: Error Analysis of Failure Cases

Error Category	Description	Example Query	Incorrect Response	Root Cause
Intent Misclassification	function manager fails to identify user's primary intent	"Mujhe kuch safe mutual funds batao aur unka expense ratio bhi."	Provides fund list but omits expense ratios	Multi-intent query handling
Factual Hallucination	Response Generator fabricates incorrect financial data	"What is the AUM of HSSC Nifty 50 fund?"	"The AUM is Rs. 500 Crores."	Lack of grounding mechanisms
Language Detection Failure	Incorrect language classification	"Ok, next."	Responds with Hindi prefix	Insufficient features in short queries
Awkward Phrasing	Grammatically correct but unnatural tone	"Is fund mein invest karna theek rahega?"	Overly formal Hindi response	Prompt engineering needs refinement

Terrence Zhao, and 1 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

Tejas Dhamecha, Rudra Murthy, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. 2021. Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8584–8595, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

IAMAI. 2024. Internet in india 2024. Technical report, Internet and Mobile Association of India (IAMAI).

Kushal Jain, Adwait Deshpande, Kumar Shridhar, Felix Laumann, and Ayushman Dash. 2020. Indic-transformers: An analysis of transformer language models for indian languages. *Preprint*, arXiv:2011.02323.

Devendra Kakwani, Kanchan Naidu, and Gayathri Band. 2025. Enhancing customer experience through ai-driven digital banking: A case study of icici bank in vidarbha. *Security Intelligence in the Age of AI: Navigating Legal and Ethical Frameworks*, page 301.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

KV Kanchan, Srihitha Patibanda, and Kalyani Gorti. Ai in banking: A case study on icici bank's ai-driven transformation.

Shailesh O Kediya, Sunita Dhote, DK Singh, VS Bidve, Shabana Pathan, and A Suchak. 2023. Are ai and chat bots services effects the psychology of users in banking services and financial sector. *Journal for ReAttach Therapy and Developmental Diversities*, 6(2):191–197.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

NSE. Report indian capital markets digital. https://nsearchives.nseindia.com/web/sites/default/files/2024-07/20240724_NSE_Report_Indian_Capital_Markets_Digital.pdf. Accessed: 2025-05-04.

Smita Ray and Anil Anirudhan. 2023. Hdfc bank's digital transformation journey. *IUP Journal of Business Strategy*, 20(4).

Kanika Sachdeva and Meenakshi Dhingra. 2024. Role of bank chatbots in managing business during crisis: A study of customers' perceptions of icici bank and sbi bank. In *Building Resilience in Global Business During Crisis*, pages 38–57. Routledge India.

Shanimon Saleem and Seena Mary Mathew. Application of artificial intelligence in banking: A study based on sbi-sia virtual assistant.

Sarvam. 2024. Sarvam 1.

Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. 2024. Hermes 3 technical report. *arXiv preprint arXiv:2408.11857*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Wenbo Zhang, Aditya Majumdar, and Amulya Yadav. 2025. Chai for llms: Improving code-mixed translation in large language models through reinforcement learning with ai feedback. *Preprint*, arXiv:2411.09073.

A Language Detection Examples

Detailed examples of language detection performance across different query types are provided to illustrate the challenges and successes of our approach.

A.1 Successful Detection Cases

Pure English: "Show me some large cap funds with high returns."

- Both Qwen2.5-0.5B and Indic-BERT correctly classify as English
- Confidence scores above 0.95 for both models

Pure Hindi: मुझे अपनी होल्डिंग्स देखना चाहता हूँ (I want to see my holdings)

- Both models correctly identify as Hindi
- Indic-BERT shows higher confidence (0.98 vs 0.91)

Code-Mixed (Hinglish): "Mere holdings mai sabse jyada returns konsa fund deta hai?" (Which fund gives the highest returns in my holdings?)

- Indic-BERT correctly identifies as Hinglish
- Qwen2.5-0.5B misclassifies as English

A.2 Challenging Cases

Financial Terminology: "Show me funds that invest in tech sector"

- Qwen2.5-0.5B incorrectly classifies as Hinglish due to pattern matching
- Indic-BERT correctly identifies as English

Short Queries: "Next" or "Ok"

- Both models struggle with insufficient context
- System defaults to previous conversation language

Mixed Script: "Mujhe HDFC Top 100 Fund का एक्सपेंस रेशियो बताओ" (Tell me the expense ratio of HDFC Top 100 Fund)

- Complex mix of Roman, English entities, and Devanagari
- Indic-BERT handles better due to multilingual pre-training