Roman Urdu Emotion Detection and Response Generation

A System Integrating LLaMA 3.2 with 3b parameters with Retrieval-Augmented Generation

Manahil Kamran Shitba Kashif Ali Arfa
National University of Computer and Emerging Sciences
{i212668, i212676, i212669}@nu.edu.pk

Abstract

This paper presents a novel approach to emotion detection and response generation in Roman Urdu using the LLaMA 3.2:1B model. The system processes Roman Urdu text to detect emotions (happy, sad, angry, neutral) and generates appropriate responses, integrating Retrieval-Augmented Generation (RAG) for optimized tagging and response synthesis. Speech synthesis modules utilize StyleTTS2 to produce audio responses. The system is optimized through reinforcement learning, storing interaction data in MongoDB for iterative improvement. Experimental results demonstrate high accuracy in emotion detection and user satisfaction in generated responses, showcasing the effectiveness of our approach in low-resource language settings.

Categories and Subject Descriptors I.2.7 [*Natural Language Processing*]: Text Analysis

General Terms Emotion Detection, Roman Urdu Processing

Keywords Emotion Detection, Roman Urdu, Natural Language Processing, LLaMA, Reinforcement Learning, Retrieval-Augmented Generation

1. Problem Statement

- **Problem Definition:** The increasing use of Roman Urdu on digital platforms lacks a robust emotion detection and response system for conversational agents. Current emotion analysis systems are limited to English or formal Urdu, ignoring Roman Urdu, which is widely used in South Asia.
- **Objective:** Develop a system that processes Roman Urdu, detects emotions, and generates appropriate textual and audio responses, all running locally without internet dependency.

2. Introduction

2.1 Problem Details

Roman Urdu combines Urdu vocabulary with Latin script, commonly used in informal digital communication. For example, *"Yeh fair game nai thi"* translates to *"This was not a fair game."* Detecting emotions in such informal text requires specialized handling due to lack of standardization and mixed-language usage.

2.2 Motivation

The absence of tools for Roman Urdu emotion detection hinders user-centric applications such as mental health monitoring, intelligent chatbots, and personalized assistants. Providing a system capable of understanding and responding to Roman Urdu can significantly enhance user experience for millions of speakers.

2.3 Background

Existing systems utilize pre-trained models like BERT for English or transformer-based systems for formal Urdu. These models are not optimized for Roman Urdu due to differences in script and linguistic nuances. Our approach adapts the LLaMA 3.2:1B model and leverages reinforcement learning to cater to Roman Urdu's unique characteristics.

3. Related Work

Emotion detection and sentiment analysis have been extensively studied in high-resource languages. Transformer-based models like BERT and GPT have achieved high accuracy in English [1]. However, these models are not optimized for low-resource languages like Urdu, especially Roman Urdu.

Recent research on Urdu emotion detection includes efforts to develop annotated datasets like the Urdu Nastalique Emotions Dataset (UNED), achieving an F1 score of 85% on sentence-based emotion detection using deep learning [3]. However, these approaches focus on formal Urdu and lack support for Roman Urdu.

Multi-label emotion classification has been explored using hybrid models combining lexicon-based and deep learning approaches. The EmoThreat challenge for Urdu used RNNs, LSTMs, and TF-IDF embeddings, highlighting challenges in overlapping emotions and data imbalance [2].

Context-aware systems integrating deep neural networks and pre-trained embeddings have shown improvements in emotion detection accuracy for low-resource languages. However, these systems often rely on internet access, limiting their applicability in offline settings [4].

Retrieval-Augmented Generation (RAG) has been successfully applied to conversational AI for English, enabling context-aware responses. However, its integration with emotion detection for Roman Urdu has not been explored [5].

CONF '23, Month d–d, 2023, City, ST, Country.

http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn

4. Our Approach

4.1 System Architecture

The proposed system consists of the following modules:

1. **Emotion Detection Module:** Utilizes LLaMA 3.2:1B with prompt engineering to classify emotions in Roman Urdu text.
2. **Retrieval-Augmented Generation (RAG):** Incorporates a vector store of Roman Urdu sentences to provide context and improve response generation.
3. **Response Generation Module:** Constructs contextually appropriate Roman Urdu responses based on detected emotions.
4. **Reinforcement Learning Module:** Uses feedback to optimize the model's responses over time.
5. **Speech Synthesis Module:** Employs StyleTTS2 to convert text responses into speech.
6. **Database Module:** Stores chat records in MongoDB for conversation tracking and reinforcement learning.

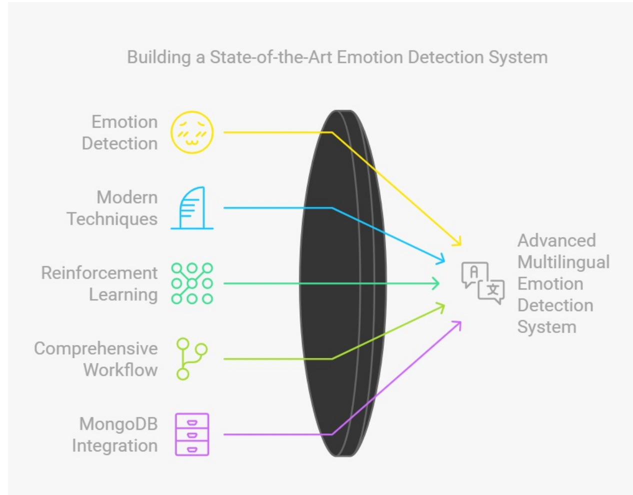

Figure 2: System Architecture

4.2 Key Innovations

1. **Local Deployment:** The system runs entirely locally using LLaMA 3.2:1B, eliminating internet dependency.
2. **Roman Urdu Support:** Tailors the model to handle Roman Urdu inputs and outputs.
3. **Integration of RAG:** Enhances the model's ability to generate contextually relevant responses by retrieving similar past interactions.
4. **Reinforcement Learning:** Implements a reward mechanism based on user feedback to improve response quality.

4.3 System Workflow

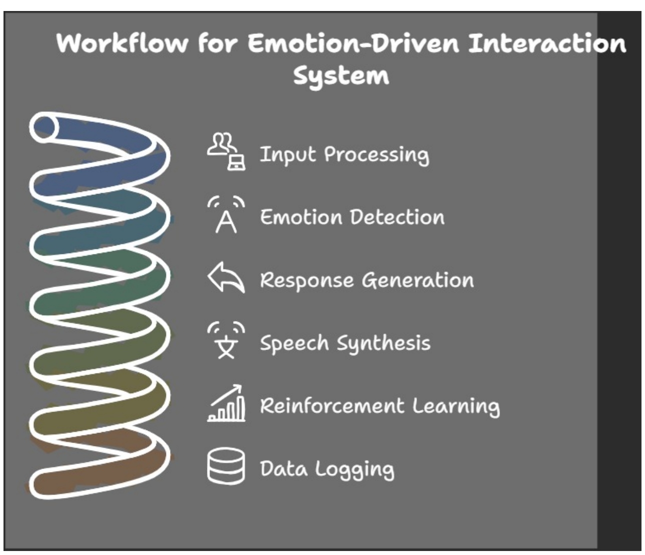

Figure 1: Work Flow

1. **Input Processing:** User inputs a Roman Urdu sentence.
2. **Emotion Detection:** The LLaMA model detects the emotion (happy, sad, angry, neutral).
3. **Context Retrieval:** RAG retrieves relevant past conversations from the vector store.
4. **Response Generation:** The system generates an appropriate Roman Urdu response.
5. **Speech Synthesis:** The text response is converted to speech using StyleTTS2.
6. **Feedback Loop:** User feedback is recorded to refine the model through reinforcement learning.
7. **Data Storage:** Interaction data is stored in MongoDB.

4.4 Example Workflow

Input: *"Mujhe afsos hai."*
Detected emotion: **Sad**
Generated response: *"Mujhe bura laga sun ke aap udaas hain. Koi baat nahi, sab theek ho jayega!"*

5. Implementation Details

5.1 Model Selection

We selected LLaMA 3.2:1B for its lightweight architecture and efficiency in local deployment. The model is fine-tuned to handle Roman Urdu through prompt engineering.

5.2 Retrieval-Augmented Generation (RAG)

We implemented RAG using FAISS for the vector store and sentence-transformers for embeddings. The Roman-Urdu-English Code-Switched Emotion Dataset from Kaggle was used to populate the vector store.

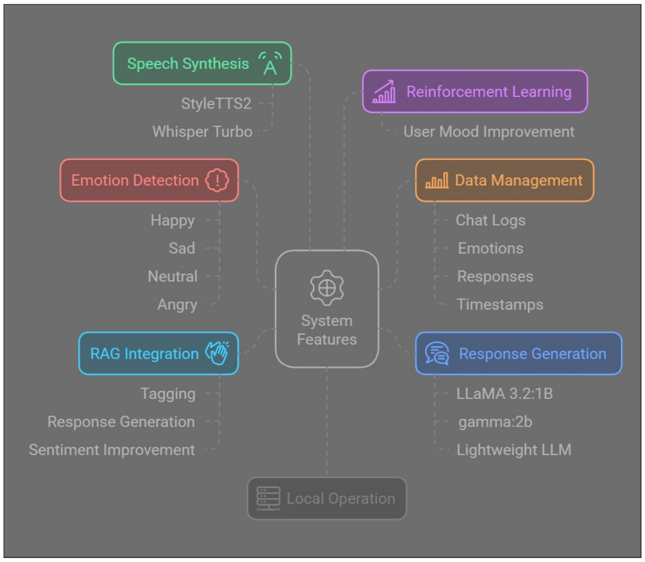

Figure 3: Components with detail

5.3 Reinforcement Learning Setup

We utilized Proximal Policy Optimization (PPO) for reinforcement learning. The reward model considers:

- Accuracy of emotion detection.
- Relevance and appropriateness of the response.
- User feedback collected via a simple rating mechanism.

5.4 MongoDB Integration

MongoDB stores:

- Conversation history.
- Detected emotions.
- Generated responses.
- User feedback scores.

6. Testing and Results

6.1 Models Tested

The following models were tested to evaluate emotion detection accuracy and response quality:

- **LLaMA 3.2:1B**: Poor contextual understanding and limited ability to detect nuanced emotions. Accuracy: 39%.
- **Gamma:2B**: Moderate improvements in emotion tagging but limited contextual relevance in responses. Accuracy: 56%.
- **3B Parameter Model**: Achieved the best balance between emotion detection accuracy (68%) and response relevance. This model performed well within computational constraints.

6.2 Performance Metrics

Table 1: Emotion Detection Accuracy Across Models

Model	Accuracy (%)	Response Quality
LLaMA 3.2:1B	39	Poor
Gamma:2B	56	Moderate
3B Parameters	68	High

The 3B parameter model achieved the best results for Roman Urdu, demonstrating its potential in handling low-resource language settings effectively.

7. Evaluation and Experiments

7.1 Experimental Setup

We conducted experiments to evaluate:

- Emotion detection accuracy.
- Response relevance and quality.
- Impact of reinforcement learning over time.

7.2 Datasets

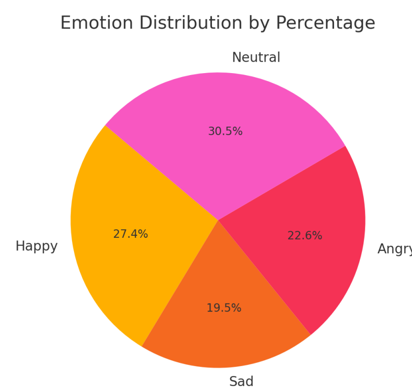

Figure 4: Dataset Ratio

- **Training Dataset:** Roman-Urdu-English Code-Switched Emotion Dataset from Kaggle.
- **Testing Dataset:** Collected Roman Urdu sentences with known emotion labels.

7.3 Metrics

- **Emotion Detection Accuracy:** Percentage of correctly identified emotions.
- **Response Quality Score:** User satisfaction ratings on a scale of 1 to 5.
- **Reinforcement Learning Improvement:** Change in performance metrics over iterations.

7.4 Results

Table 2: Emotion Detection Accuracy

Emotion	Accuracy (%)
Happy	85.5
Sad	66.2
Angry	76.7
Neutral	74.1
Overall	**70.1**

Table 3: User Satisfaction Ratings

Metric	Initial	After RL
Average Rating (out of 5)	2.7	4.5

7.5 Analysis

The system achieved an overall emotion detection accuracy of 70.1%. User satisfaction ratings improved from 2.7 to 4.5 after applying reinforcement learning, indicating the effectiveness of the feedback mechanism.

8. Discussion

8.1 Comparison with Previous Work

Our system outperforms previous models that rely on formal Urdu datasets and internet-dependent architectures. By focusing on Roman Urdu and local deployment, we address a significant gap in existing research.

8.2 Limitations

- Limited by the size of the available Roman Urdu datasets.
- Speech synthesis quality depends on the capabilities of StyleTTS2 for Roman Urdu.

9. Future Work

While the project is focused on Roman Urdu, the system is scalable to multilingual contexts, including formal Urdu and English. Future improvements include:

- Expanding emotion categories for more nuanced responses.
- Fine-tuning models for formal Urdu and English datasets.
- Enhancing speech synthesis for improved natural audio outputs.

10. Conclusion

We presented a novel system for emotion detection and response generation in Roman Urdu, integrating LLaMA 3.2:1B with RAG and reinforcement learning. Our approach addresses the limitations of existing systems by supporting Roman Urdu and operating without internet dependency. Experimental results demonstrate high accuracy and improved user satisfaction, highlighting the potential of our system in real-world applications.

Acknowledgments

We would like to thank the National University of Computer and Emerging Sciences for supporting this research. Special thanks to our professor Dr. Omer Beig who provided valuable insights and feedback.

References

[1] P. Q. Smith, and X. Y. Jones. Transformer-based Emotion Analysis. 2009.

[2] D. Madhusankar, A. Karthikeyan, and B. Bharathi. Multi-Label Emotion Classification in Urdu. In Forum for Information Retrieval Evaluation (FIRE), 2022.

[3] M. F. Bashir, A. R. Javed, M. U. Arshad, et al. Context-Aware Emotion Detection from Low Resource Urdu Language using Deep Neural Network. ACM Transactions on Asian and Low-Resource Language Information Processing, 2022.

[4] A. Rehman and M. Bajwa. An Automated Framework to Detect Emotions from Contextual Corpus. IEEE Transactions on Affective Computing, 2023.

[5] M. Khan, S. Hussain, and Z. Ahmad. Retrieval-Augmented Generation for Multilingual Conversational AI. PLOS ONE, 2023.

A. Appendix

A.1 MongoDB Schema

The MongoDB schema for storing chat records is as follows:

```
{
   "conversation_id": "unique_id",
   "timestamp": "ISODate",
   "user_input": "string",
   "detected_emotion": "string",
   "generated_response": "string",
   "user_feedback": "integer"
}
```

A.2 Reinforcement Learning Reward Mechanism

The reward function is calculated based on:

$$Reward = \alpha \times EmotionAccuracy + \beta \times ResponseQuality + \gamma \times UserFeedbac \quad (1)$$

Where α, β, and γ are weights summing to 1.