Emotion-Aware Chatbot: Technical Documentation

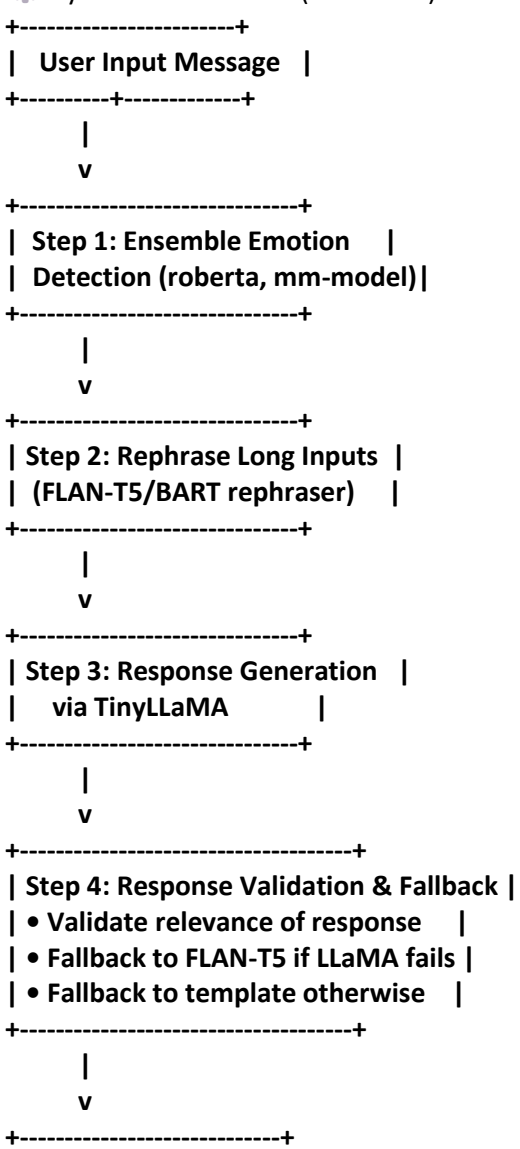
🔍 Problem Statement

To develop an emotion-aware chatbot capable of understanding the user's emotions from input text and generating empathetic responses. The solution should be lightweight and executable on Kaggle or local machines without GPU acceleration.

🎓 Approach Overview

We designed a multi-stage pipeline that integrates **ensemble-based emotion classification**, **response generation via TinyLLaMA and Flan-T5**, and **fallback mechanisms** to ensure robustness and empathy.

⚙️ System Architecture (Flowchart)



| Final Emotion + Response |
+----------------------------+

Dataset

We used the [dair-ai/emotion](https://huggingface.co/datasets/dair-ai/emotion) dataset from Hugging Face:

- 6 emotion classes: joy, sadness, anger, fear, surprise, love
- Used ensemble models (Multilingual MiniLM + RoBERTa) for robust emotion classification.

🔝 **Emotion Classification (Ensemble)**

- Models: roberta-base, microsoft/deberta-v3, and multilingual-MiniLM.
- Ensemble Strategy: Average softmax scores from multiple classifiers.
- Output: Predicted emotion from the label set: {joy, sadness, anger, fear, surprise, love, neutral, excitement}

🖍️ **Input Preprocessing**

- Rephraser: If input > 20 words, rephrase using Flan-T5 or BART for better understanding by classifiers or LLM.
- Validator: Ensures rephrased sentence preserves meaning and grammar.

🧠 **Empathetic Response Generation**

Primary Generator:

- TinyLLaMA with emotion-conditioned prompt.
- Prompt format:

"You are an empathetic chatbot. The user is feeling {emotion}. They said: '{input}'. Reply kindly and naturally."

Fallback 1:

- If LLaMA response is too short or hallucinates emotion → switch to Flan-T5 with prompt:

"Respond empathetically to this message knowing the user feels {emotion}: {input}"

Fallback 2:

- If both LLMs fail → use template_responses[emotion] dictionary.

⚖️ Response Validation Logic

```
if len(response) < 5 or "reply kindly" in response:
    fallback = "FLAN"
elif top_emotion not in response.lower() and not validate_response_with_emotion(response, top_emotion):
    fallback = "Template"
else:
    fallback = "LLaMA"
```

📝 Template Responses (Last Resort)

{

"joy": "That's wonderful to hear! 😊 ",

"sadness": "I'm really sorry you're feeling this way. You're not alone — I'm here for you. 💙 ",

"anger": "I understand you're upset. It's okay to feel this way — want to talk about it?",

"fear": "It's okay to feel overwhelmed sometimes. Take a breath — you're doing your best. 🧘 ",

"surprise": "Wow, that does sound unexpected. Thanks for sharing it with me!",

"love": "That's so heartwarming. It's beautiful to feel this way. 💖 ",

"neutral": "Thanks for sharing that. I'm listening. 🤝 ",

"excitement": "That's amazing! 🎉 So happy for you!"

}

🚫 Limitations

- **Model Size & Capability:**
    - TinyLLaMA and Flan-T5-small are lightweight but not deeply nuanced.
    - Context length and abstraction understanding are limited in long/complex emotional narratives.

- **Hardware Constraints:**
    - Designed to work without GPU — on Kaggle CPU and local machines.
    - Larger models (Zephyr, Mistral) not used due to RAM/runtime limits.

🚀 Future Improvements

- Integrate **Retrieval-Augmented Generation (RAG)** for contextually rich empathetic responses.
- Fine-tune LLaMA or use **Zephyr 7B/DialoGPT** as scalable response generators.
- Add multilingual support for global user base.
- Train emotion classifier on longer context-rich emotion datasets.