

DataDecode : Emotion Detection And Chat Analyzer In Chat Application

Tejas Bhalla, Ishita, Priya Sharma Harit
Vivekananda Institute of Professional
Studies - Technical Campus
Delhi, India

Abstract

This report presents the development and implementation of real-time chat applications with the help of an integrated emotional recognition system. The main goal of this project was to create a platform that not only promotes immediate messaging between users, but also provides valuable insight into the emotional nuances of the conversation. The frontend for the application was created with flutter and darts in an Android studio development environment. This was chosen for cross-platform functions and efficient UI rendering. FireBase was used as a back-end infrastructure to manage real-time, user authentication, and news storage synchronization to ensure a robust and scalable communications framework. This model uses bi-directional encoder displays from the latest technology (NLP) TransArchitecture (Trans), Natural Language Processing (NLP) to identify and classify underlying emotions. To bridge the gap between mobile applications and emotion detection models, a relaxed API has been designed and implemented to enable seamless and efficient data transmission and processing. Successful integration of these technologies means real-time exchange of chat applications, immediate feedback to promote emotional communication tones for users, improved general interactive experiences and fostering online interactions. The report lists structural considerations, development methods, and reviews of system performance and accuracy.

Keywords: real-time chat, emotion detection, NLP, machine learning, LSTM, Flutter, Firebase, mobile computing

Introduction

This section introduces the core concepts and motivations for developing real-time chat applications and integrates them into emotion detection. Real-time chat platforms have become an essential tool in many fields, from personal communication to professional collaboration. Despite their widespread use, traditional messaging systems are often ignored when communicating or interpreting the emotional context of a conversation, which can lead to weak misunderstandings and connections. The project aims to address this limitation by embedding detection functions emotions into a chat interface and enriching digital interactions with emotional intelligence.

The development of the application relied on several key technologies and tools:

- **Flutter and Dart:** Google's UI toolkit Flutter was chosen to create natively compiled applications for both iOS and Android from a single codebase. Dart, a programming language used in Flutter, supports efficient and reactive UI development. Android Studio was used as the most important integrated development environment (IDE) for coding, debugging and testing.
- **Firebase:** We used Firebase, Google's cloud-based platform, to handle backend functions. FireBase Real-Time Database and FireBase Cloud Messaging have activated real-time synchronization and news delivery, while FireBase authentication has managed secure user registration and identity services. Together, these tools provided a scalable and reliable infrastructure that simplifies back-end development.

- **Python:** Python was chosen for developing the emotion detection model due to its extensive ecosystem of libraries in machine learning and natural language processing (NLP). Libraries such as TensorFlow and Transformers made it well-suited for building advanced NLP models.
- **BERT (Bidirectional Encoder Representations from Transformers):** The BERT model was employed as the core of the emotion recognition system. Known for its state-of-the-art performance in NLP tasks, BERT excels in understanding context and linguistic nuance, making it ideal for accurately identifying the emotional tone in text-based messages.
- **Application Programming Interface (API):** A RESTful API was developed to facilitate communication between the Flutter front-end and the Python-based emotion detection model. This interface enables seamless data exchange, allowing the application to send user messages for analysis and receive emotion predictions in real-time.

communication. Joshi (2019) analyzed WhatsApp group chats to detect group sentiment trends using R, while Zou et al. (2015) explored the use of bidirectional LSTM for sentiment classification in microblogging texts.

Deep learning models, especially LSTM and CNN-LSTM hybrids, have shown superior performance in modeling sequential data and understanding context in language. Compared to traditional ML models like Naïve Bayes or SVM, LSTMs handle long-term dependencies better, a critical feature when analyzing conversational text.

Additionally, research by Bird et al. (2009) and Mitchell (1997) laid foundational work in NLP and machine learning, respectively. These works emphasized the potential of combining linguistic features with learning models to interpret user intent and sentiment.

The integration of real-time processing with emotional analysis remains a developing field, and this work contributes by combining robust NLP methods with mobile computing technologies to deliver on-device predictions.

The integration of these technologies leads to systems that can improve user communication in a variety of impressive ways. By providing insight into the emotional tone of the message, the application promotes it more sensitive and thoughtful conversation. It also helps to clarify potentially vague messages and reduce the risk of misunderstanding. Furthermore, such emotionally conscious systems promise a more comprehensive application, including mental health support. This application analyzes the key moods and is important in real time in a customer service environment, where understanding of emotional information is important.

Literature Review

Several studies have addressed sentiment analysis in social media and chat-based