

MoodSense: A Python-Based Emotion Detection Chatbot for Personalized Entertainment Recommendations

Supriya Lole¹, Janhavi Mali², Suyog patil³

¹ Department of MCA, G H Raison College of Engineering and Management, Wagholi, Pune, Maharashtra, India

² Department of MCA, G H Raison College of Engineering and Management, Wagholi, Pune, Maharashtra, India

³ Department of MCA, G H Raison College of Engineering and Management, Wagholi, Pune, Maharashtra, India

ABSTRACT

In recent years, conversational artificial intelligence (AI) has evolved from simple rule-based systems to interactive assistants capable of understanding user context and emotions. However, many chatbot frameworks rely on complex machine learning models and cloud infrastructure, making them resource-intensive for lightweight applications.

This study proposes MoodMate, a Python-based mood-aware chatbot that provides personalized recommendations for movies, music, and motivational content. The system uses a keyword-based sentiment detection method to identify emotions such as happiness, sadness, or boredom.

1. INTRODUCTION

MoodSense is a Python-based emotion detection chatbot designed to recognize user moods and provide personalized entertainment recommendations. In today's fast-paced digital world, people often look for movies, music, or inspirational content that matches their emotions. However, finding suitable entertainment from countless online options can be difficult.

MoodSense addresses this challenge by creating a conversational system that understands emotional cues and suggests relevant content accordingly.

The chatbot works using a rule-based emotion detection approach where user input is analyzed for mood-related keywords such as happiness, sadness, stress, or boredom. Based on the detected emotion, MoodSense recommends

The modular architecture ensures efficient performance with low computational requirements and fast response times. The proposed system demonstrates how lightweight conversational AI can deliver personalized experiences while remaining accessible and scalable for educational and wellness applications.

KEYWORDS

Chatbot, Python, Conversational AI, Mood Detection, Recommendation System, API Integration.

suitable entertainment options including movies, music playlists, or motivational quotes. To provide real-time and dynamic suggestions, the system integrates multiple APIs such as TMDb for movie recommendations, Spotify for music suggestions, ZenQuotes for inspirational quotes, and OpenWeatherMap to consider weather conditions while suggesting content.

Technically, MoodSense follows a modular architecture that includes input processing, emotion detection, API communication, and response generation. Python's requests library is used to fetch data from external APIs, and the results are presented to users through friendly conversational responses. Overall, MoodSense demonstrates how lightweight conversational systems can deliver emotionally aware interactions, enhance user engagement, and provide personalized digital entertainment experiences.

detection and sentiment analysis to create more empathetic and personalized conversational experiences. Sentiment analysis techniques allow chatbots to identify emotional cues from textual inputs and respond appropriately, which significantly improves user engagement and satisfaction.

Several studies have proposed emotion-aware chatbots capable of adapting their responses based on users' feelings. For example, research on an AI chatbot with real-time

2. LITERATURE REVIEW

Conversational chatbots have become an important component of modern Human-Computer Interaction (HCI), enabling systems to communicate with users in a natural and interactive manner. Early chatbot systems mainly focused on answering queries or providing customer support. However, recent research has explored the integration of emotion

emotion sensing demonstrated that combining natural language processing with emotion recognition techniques can significantly enhance user interaction and emotional connection with the system. The study reported improved engagement and satisfaction when chatbots dynamically adjusted responses according to detected emotional states.

Other researchers have explored recommendation-based chatbots that provide entertainment suggestions such as movies or music according to user preferences. A survey on intelligent chatbots for entertainment recommendation highlights that chatbots can reduce information overload by filtering large amounts of online content and providing personalized suggestions based on user interests and emotions.

Emotion recognition techniques in chatbots have also been implemented using advanced machine learning models. Some systems utilize deep learning approaches such as CNN, LSTM, and BERT embeddings to detect emotions from textual data and recommend suitable music or other content aligned with the detected mood. These systems demonstrate the potential of combining emotion detection with recommendation mechanisms to create emotionally intelligent digital assistants.

Despite these advancements, many systems rely on complex machine learning models and high computational resources. Therefore, lightweight solutions using rule-based logic and API-driven architectures are gaining attention. Systems like MoodSense aim to bridge this gap by implementing mood detection and personalized entertainment recommendations through simple yet effective Python-based conversational frameworks.

3. METHODOLOGY

3.1 User Input Processing

The first stage of the system involves collecting and processing user input. MoodSense interacts with users through a text-based interface where individuals can type messages describing their feelings, mood, or general thoughts.

When the user sends a message, the system first performs input preprocessing. This includes cleaning the text by removing unnecessary characters, converting the text into lowercase, and identifying important words. This step ensures that the system can easily analyze the message and detect emotional indicators.

The processed text is then prepared for the emotion detection stage. Proper input processing improves the accuracy of mood identification and ensures smoother interaction between the user and the chatbot.

3.2 Emotion Detection Using Keyword-Based Analysis

Emotion detection is the core component of the MoodSense chatbot. In this stage, the system analyzes the processed user input to determine the emotional state of the user. MoodSense uses a keyword-based rule approach to identify

emotions such as:

- Happy
- Sad
- Stressed
- Bored
- Excited

The system maintains a predefined list of keywords associated with different emotions. When a user message is received, the chatbot compares the words in the message with the emotion keyword database. If matching keywords are detected, the system categorizes the user's emotional state accordingly.

This lightweight method allows the chatbot to quickly detect mood patterns without requiring complex machine learning models.

3.3 Personalized Entertainment Recommendation

After detecting the user's emotion, the system generates personalized entertainment suggestions. The goal is to provide content that matches or improves the user's emotional state.

MoodSense integrates multiple external APIs to fetch real-time recommendations:

- TMDb API – Provides movie recommendations.
- Spotify API – Suggests music playlists based on mood.
- ZenQuotes API – Generates motivational or inspirational quotes.
- OpenWeatherMap API – Retrieves weather data for contextual suggestions.

For example:

- If the user feels sad, the system may recommend uplifting music or motivational quotes.
- If the user feels bored, it may suggest trending movies or energetic playlists.
- If the weather is rainy, the chatbot may suggest relaxing movies or calm music.

This integration allows MoodSense to deliver dynamic and relevant recommendations.

3.4 Response Generation and Conversation Flow

Once the recommendation data is retrieved, the system converts it into natural conversational responses. Instead of displaying raw data, the chatbot presents suggestions in a friendly and engaging format.

For example, the chatbot may respond with messages such as:

- "It looks like you're feeling a bit stressed. Here's a calming playlist you might enjoy."
- "Since you're feeling bored, here's a popular movie you could watch."

Predefined response templates are used to maintain a natural conversation flow. This makes the chatbot interaction feel

more human-like and enjoyable.

3.5 Feedback and System Improvement

To improve the user experience, MoodSense also includes a feedback mechanism. After providing a recommendation, the chatbot allows users to request alternative suggestions if they are not satisfied.

For example, users can respond with phrases such as:

- “Suggest another movie.”

4. RESULTS AND DISCUSSION

The proposed system **MoodSense: A Python-Based Emotion Detection Chatbot for Personalized Entertainment Recommendations** was implemented and tested in a Python-based conversational environment. The primary objective of testing was to evaluate whether the chatbot correctly detects user emotions and provides relevant entertainment recommendations such as movies, music, and motivational quotes.

The evaluation involved testing different interaction scenarios including various emotional inputs, API response handling, recommendation accuracy, and overall system performance. Users were asked to interact with the chatbot using different mood expressions such as happiness, sadness, boredom, and stress. The system responses were then analyzed to determine whether the recommendations matched the detected emotions.

4.1 System Observations

The first important observation relates to **emotion detection accuracy**. The chatbot successfully identified user moods based on predefined keyword matching. When users entered statements such as “I feel sad today” or “I am very bored,” the system correctly classified the emotional state and generated appropriate responses.

The system also demonstrated effective **personalized recommendation capability**. After detecting the emotion, the chatbot retrieved suitable suggestions through integrated APIs.

For example:

- When users expressed **sadness**, the chatbot suggested motivational quotes and calming music.
- When users expressed **boredom**, the chatbot recommended trending movies or energetic playlists.
- When users expressed **happiness**, the chatbot suggested upbeat music or popular entertainment options.

The integration of external APIs such as movie, music, and quote services worked efficiently. The chatbot was able to fetch real-time data and convert it into conversational responses.

The **weather-awareness feature** also worked effectively. For example:

- During rainy weather, the system recommended relaxing movies or calm music.
- During sunny weather, it suggested energetic playlists or

- “Give me more music options.”

The system then retrieves additional recommendations from the APIs. This feature ensures flexibility and keeps the interaction engaging.

Future improvements may include advanced features such as:

- Machine learning-based emotion detection
- Voice-based interaction
- User preference tracking for better personalization
- Integration with additional entertainment platforms

outdoor-themed content.

These observations confirm that the chatbot successfully combines emotional analysis with contextual information to enhance the recommendation experience.

4.2 Limitations

Although the MoodSense system performs effectively, several limitations were identified during testing. One limitation is related to the **keyword-based emotion detection approach**. Since the system relies on predefined keywords, it may not correctly interpret complex sentences, sarcasm, or mixed emotions expressed by users.

Another limitation involves **dependency on external APIs**. If an API service is unavailable or slow to respond, the chatbot may take longer to generate recommendations or may temporarily fail to provide suggestions.

The system currently supports a **limited number of emotion categories**, which may restrict its ability to fully understand nuanced human emotions. Additionally, the chatbot mainly processes text-based input and does not yet support voice input or advanced natural language understanding techniques. Despite these limitations, the system still provides a useful and interactive experience for basic emotional recognition and recommendation tasks.

4.3 Accuracy and Performance

The MoodSense chatbot was evaluated to measure both **emotion detection accuracy** and **system performance**. During testing, the system correctly detected emotional keywords in most user inputs and generated relevant responses.

The chatbot responded quickly because it uses a **lightweight rule-based approach** instead of computationally intensive machine learning models. This ensured fast processing of user inputs and smooth conversational flow.

The use of API-based recommendations also improved the system’s ability to provide **real-time and dynamic entertainment suggestions**. In most cases, recommendations were generated within a few seconds, which maintained a responsive user experience.

Overall, the system demonstrated stable performance under normal usage conditions. The lightweight architecture ensures that the chatbot can operate efficiently even on systems with limited computational resources.

4.4 Future Work

The MoodSense system can be improved by incorporating more advanced technologies to enhance emotion detection and personalization capabilities. One possible improvement is the use of **machine learning or deep learning models** such as Natural Language Processing (NLP) techniques to detect emotions more accurately from complex user sentences.

Another potential enhancement is the integration of **voice-based interaction**, allowing users to communicate with the chatbot using speech instead of text. This would make the

system more interactive and accessible.

The system can also be expanded by including additional **entertainment platforms and recommendation sources**, enabling a wider variety of suggestions such as podcasts, books, or games.

Furthermore, implementing **user preference learning** could help the chatbot remember previous interactions and provide more personalized recommendations over time.

These improvements would make MoodSense more intelligent, adaptive, and capable of delivering richer emotionally aware conversational experiences

5. CONCLUSION

In this research, an emotion-aware conversational system MoodSense: A Python-Based Emotion Detection Chatbot for Personalized Entertainment Recommendations has been designed and explained. The main objective of this system is to detect the emotional state of users and provide suitable entertainment suggestions that can improve their mood and enhance user engagement.

The system analyzes user messages and identifies emotions using a keyword-based emotion detection approach. Instead of storing complex models, the chatbot processes the user input through predefined emotion categories such as happiness, sadness, boredom, and stress. This lightweight method allows the system to quickly recognize the user's emotional state and generate appropriate responses.

The chatbot integrates several external APIs to provide dynamic recommendations. Movie suggestions are obtained through the TMDb API, music recommendations are generated using the Spotify API, and motivational quotes are retrieved from the ZenQuotes API. Additionally, weather data from the OpenWeatherMap API helps the chatbot provide context-aware suggestions based on environmental conditions.

The testing results show that the system responds effectively to user inputs and provides relevant entertainment recommendations. MoodSense demonstrated stable performance and quick response time during interactions. Although the keyword-based emotion detection approach has certain limitations in understanding complex

language expressions, the system still offers a practical and engaging conversational experience.

In conclusion, MoodSense demonstrates how a Python-based chatbot can provide personalized entertainment recommendations through simple emotion detection techniques. The system highlights the potential of conversational AI to create interactive, mood-aware digital assistants that enhance user experience and digital well-being.

6. References

- [1] Jurafsky Daniel and James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education, 2009.
- [2] Python Software Foundation. Python Language Reference Manual. Available: <https://www.python.org>
- [3] Spotify for Developers. Spotify Web API Documentation. Available: <https://developer.spotify.com/documentation/web-api/>
- [4] The Movie Database. TMDb API Documentation. Available: <https://developer.themoviedb.org>
- [5] OpenWeather. OpenWeatherMap API Guide. Available: <https://openweathermap.org/api>
- [6] Requests Python Library. Requests: HTTP for Humans Documentation. Available: <https://docs.python-requests.org/>
- [7] BeautifulSoup Library Documentation. Beautiful Soup for Web Scraping. Available: <https://www.crummy.com/software/BeautifulSoup/>