International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Emotion-Aware Conversational AI: A Chatbot Song Recommender System Utilizing Natural Language Processing and External APIs

N Advytha Reddy[1], Harshith Chowdary Y[2], Jai Ganesh[3], Ms T Lahari[4]

[1,2,3,4]Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, India

Abstract:
The swift progress of Artificial Intelligence (AI) and Machine Learning (ML) has made it possible to create intelligent conversation agents that not only understand but also react to human feelings. In this context, the project presents the Chatbot Song Recommender System which is a smart integration of Natural Language Processing (NLP), sentiment analysis, and music recommendation algorithms. To some extent, this system even supersedes the traditional chatbots deployed in business as it is aimed at casual interaction and human-like entertainment personalized for the particular user. The IBM Tone Analyzer API is used by the chatbot to perform the analysis of the user's emotional tone which could be joy, sadness, or anger, for instance, during the talk. The emotion detected is then used by the system to pull the relevant songs from the Last.fm API that would be in sync with the user's mood. The development of the chatbot is done with Python, while scikit-learn and TensorFlow are among the libraries used for implementing the machine learning functionalities. This method intends to make the user experience more interactive, thereby connecting conversational AI and emotional intelligence, as well as revealing the possibilities of AI in the realm of interactive entertainment applications.

Keywords: Chatbot, Song Recommender System, Natural Language Processing (NLP), Sentiment Analysis, Emotion Detec- tion, IBM Tone Analyzer API, Last.fm API, Conversational AI.

I. INTRODUCTION

In the last few years, the technologies of Artificial In- telligence (AI) and Machine Learning (ML) have completely revolutionized the human-technology interaction. One of the most talked about innovations is the introduction of chatbots, which are capable of human-like conversations and being a user's assistant in real-time across several platforms. These systems powered by AI have already been mixed with the human's daily life and are making digital communication more accessible and responsive.

Nevertheless, the majority of the current chatbots are still mainly focused on business and they provide only limited emotional understanding besides transactional support. This very fact can lead to very uncompromising and mechanistic user experiences, especially when empathy and personalized interaction are demanded. Gradually the users are wanting the systems that can mood-swing, and adjust their responses, thus the emergence of emotionally intelligent chatbots. The project at hand aims at covering this abyss by putting forward a Chatbot Song Recommender System that would be utilizing Natural Language Processing (NLP) and sentiment analysis as tools to be able to process the emotional tone of

the user throughout the conversation. After establishing the user's state of mind, the system would suggest music that resonates with the user's mood, which in turn would make the communication both significant and fun.

The implementation of the system is done through the use of state-of-the-art APIs such as the IBM Tone Analyzer for emotion detection, Last.fm for music recommendations, and also Python-based libraries like TensorFlow and Scikit-learn for robust applications. The project has proven the viability of amalgamating AI technologies for the development of smart, user-oriented applications that facilitate daily digital interaction. The end product of the system is a remarkable combination of conversational AI and emotion-driven mu- sic recommendation, which provides a more interactive and human-like user experience; in turn, this has a positive effect on user engagement.

A. Demerits of the Existing System

Although the existing chatbot and music recommendation systems exhibit a high level of technical expertise, they still have some major drawbacks that make them less effective in giving a truly personalized experience. One of the major problems is that most chatbot systems still lack emotional intelligence and are mainly designed to operate either on pre-set responses or apply rule-based logic. Thus, they can hardly acknowledge the emotional tone or react accordingly to a user's interaction which in turn makes the conversation feel mechanical and unconnected to the user's mind state. In addition, the current digital music platforms still do not take into account the real-time feelings of the user and thus, they rely on past listening habits and data for making suggestions which mean the user has no emotional connection with the suggestions. Hence, the suggestions may be similar to past ones made, and be quite general as well as not providing the user with a very personalized alignment of the current mental or emotional state.

B. Proposed System and Objectives

The suggested Chatbot Song Recommender System is aimed at surmounting these drawbacks by merging natural language dialogue with instantaneous emotional intelligence. The system chats with the user informally, uses the IBM Tone Analyzer API to detect the emotional tone of their texts, and creates song recommendations in an appropriate manner by utilizing the Last.fm API.

The main aims of this project are as follows:

1. To create an emotionally intelligent chatbot that can comprehend and reply like a human to user input, as well as to point out the users' emotions through the use of NLP techniques.
2. To use the IBM Tone Analyzer API for real-time sen- timent analysis to determine the user's emotional tone during live chat interactions.
3. To suggest songs based on the user's mood by connect- ing to the Last.fm API, which is an enhancement of real-time personalization.
4. To create a single system that combines the capabilities of conversational AI and recommendation and provides the user with an intriguing experience.

II. LITERATURE SURVEY

With the introduction of integrated systems that can deal with user emotions in real-time, the domain of conversational AI and music recommendation has rapidly evolved. The initial stages of the development of AI-based standard Chatbot Music Recommendation Systems were aimed at user engagement through personalized music suggestions [1]. These systems usually employed techniques such as passive recommendations based on contextual factors like time of day and weather conditions and active

suggestions during user interaction [2]. By incorporating real-time data from external APIs and using these systems tried to language processing (NLP), natural improve recommendations over time [2,3].

A large amount of energy has been put into the construction of Chatbot Music Recommender Systems according to user emotions [4]. These systems generally have machine learning algorithms like collaborative filtering and content-based filter- ing in the background to provide personalized recommenda- tions [280]. The process, algorithms, and system flow give importance to the use of sentiment analysis for the purpose of improving user interaction and music recommendation, at times making use of APIs like Spotify or Last.fm [5,6]. The purpose remains to make an emotion-based music recom- mendation system through a chatbot interface that suggests personalized music recommendations based on the moods reported via chat [6]. This method supports communication in the natural language form which makes the whole process of referral more engaging and interactive [7].

The use of multimodal approaches for emotion recognition has been a topic of research in more advanced studies [8]. For example, these systems deploy intricate neural networks such as Convolutional Neural Networks (CNN), Long Short- Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and LSTM-GRU networks with BERT embeddings to detect emo- tions in different input modes like text, webcam facial expressions, or audio [8,9]. In addition to using the Kaggle Emotion Dataset for training, these systems also create web-based frameworks to perform real-time emotion recognition [10].

The ultimate goal of these studies is to develop a personalized music recommendation system that suggests tracks in line with the user's emotional state as per the identified emotion .

Moreover, the researchers are working on the emotion-based music playlist recommendation systems which are interactive chatbots to judge the user's emotional state through human-like conversations [10,11]. The mood considered is derived from the sentiment and the system computes a mood score and creates a playlist of music using the Spotify API and services like it . The approach has the benefit of giving more attention to the user mood over traditional methods based on historical or popularity data, thus changing the music discovery experience considerably [12].

III. METHODOLOGY AND IMPLEMENTATION

A. Methodology

The primary approach of the Chatbot Song Recommender System consists of combining natural language processing (NLP), sentiment analysis, and external music APIs to provide personalized song selections via user's dialogue mood. The system's operation chain gets initiated when the user talks through the interface. The textual input is subjected to NLP processing so that the emotional context could be recognized. The conversational data is then transferred to the IBM Tone Analyzer API, which detects emotional signals like joy, sadness, anger, and calmness. The emotional labels are converted into a mood score that acts as the foundation for picking up songs one by one.

The main programming language for the system is Python because of the wide range of libraries for machine learning and chatbot development that belong to its ecosystem, e.g., TensorFlow, scikit-learn, and NLTK. After determining the mood score, it is sent to the music recommendation system. This system communicates with the Last.fm API, which provides song metadata according to emotion-related attributes being the basis for the retrieval. A matching algorithm then selects and orders the songs that correspond best to the user's mood, and the stack is finally shown through the chatbot. At the same time, the backend performs the user data, conversation logs, and API integrations matters using Django

or ASP.NET frame- works, while the frontend provides a smooth user experience employing Flutter or React Native platforms.

B. System Architecture

The system design is based on a modular architecture, so it can be adjusted and repaired easily. The user interface (UI) is the first point of interaction for the text flow. The input is sent to the frontend intercess and the NLP processor does the processing. The information is then sent to the backend chatbot engine, which controls the conversation and starts analyzing the tone. The system uses external APIs for two important functions: emotion detection (through IBM Tone Analyzer) and song retrieval (through Last.fm API). The Music Recommendation Engine gets the emotional score and connects with the Song Database/API Handler to get tracks that match the mood. The chatbot engine prepares the final answer that includes the song recommendations and sends it back to the user.

Fig. 1. System Architecture of the Chatbot Song Recommender System

C. Algorithm and Workflow

The procedure implemented by the system is a sequential algorithm that goes from user initiation to song delivery. First, the algorithm initializes the chatbot interface and is on standby for the user to input. The text input from the user is received and forwarded to the emotion analysis module, which uses the IBM Tone Analyzer API to return emotional tones (e.g., joy, sadness, anger).

After identifying the emotion, the system correlates that emotion to a certain music mood category. For instance, joy is associated with upbeat songs while sadness is linked with calming or relaxing songs. The chatbot then Queries the Last.fm API for songs corresponding to the mapped mood category and assembles a list of recommendations. The chatbot interface displays the suggested songs to the user. The conversational loop is maintained allowing the user to either ask for more suggestions or end the session.

TABLE I
TECHNOLOGY STACK OVERVIEW

Component	Technology Used
Language	Python
Backend	Django / ASP.NET
Frontend	Flutter / React Native / Ionic
NLP / Tone Analysis	IBM Tone Analyzer API
Music API	Last.fm API
ML Libraries	scikit-learn, TensorFlow
Database	MySQL Server
Design Tools	Use Case Diagrams, Flow Charts, ER Diagrams

IV. RESULTS AND DISCUSSION

The Chatbot Song Recommender System was a great success when it came to evaluation and also displayed as a remarkable combination of conversational AI and real- time emotion detection giving personalized entertainment. The backend of the system uses Python with Django, and the frontend is built using frameworks like Flutter or React Native. The main APIs incorporated were the IBM Tone Analyzer for advanced emotion detection and the Last.fm API for lively music suggesting..

A. User Interface and Interaction

The landing page of the site introduces the visitors to a stylish and interactive model. The ”VibeCheck Tunes” plat-form is the medium of interaction, and its attributes – a smooth interface and a cordial chatbot persona – are easily perceived. The chat window is the most prominent area where the users feel like right away chatting and are even encouraged to tell about their feelings or thoughts. This easy-going and unprofessional manner of conversation plays a significant role in getting the natural emotional input that is appropriate for the tone analysis.

The user interface has sorted the results neatly, giving the user the option to select from different mood playlists like Happy Vibes, Chill Sad, and Energy Boost which makes it a very simple process of discovering music. The songs on the playlist page are nicely arranged under the respective albums and the play buttons along with the clear typography ensure that the music browsing experience is really enjoyable.

Fig. 2. Coverpage and Initial Chat Interaction of the System (VibeCheck Tunes)

B. Performance and Testing

The system’s performance was assessed through both its functional and non-functional requirements, with accuracy, responsiveness, and reliability being the primary focus. Among the non-functional requirements, the specification of a re- sponse time for emotion detection and song recommendation was set to be less than two seconds which is of utmost importance for a smooth conversational interaction. Several key test cases confirmed the existence of main features: • User Registration and Authentication were these functionalities through successful creation.

B. Future Scope

In addition to the above, the future improvement of the sys- tem could also consist of enlarging its scope and intelligence:

1. Multilingual and Regional Language Support: The chatbot can be equipped with multilingual capabilities that will allow it to support major international and regional languages, thereby attracting a larger audiene, particularly in diverse nations like India. The utilization of Natural Language Processing (NLP) models trained in regional languages such as Hindi, Tamil, and Bengali will not only make the system accessible but also more inclusive.
2. Personalized Mood Journals and Analytics: The chat- bot can become a personal mood buddy by keeping a mood diary for the users. The users can be enlightened about their emotional state with the help of visual mood graphs and sentiment trends over time which may lead to self-awareness and even early mental health support.
3. AI-Based Voice Interaction: The implementation of advanced voice-based interaction via AI-enabled voice recognition will greatly enhance the naturalness of the conversation flow. The users will be able to talk with the chatbot as if they were talking with a person, thus the system would be more friendly, especially to the less technically inclined users.
4. Cross-Platform Integration: The system is designed to be user-friendly and available on different platforms like WhatsApp, Discord, Telegram and also can be integrated into home automation

systems. Users will not be limited to one platform for their music recommendations and that is a great way to increase the system's reach and flexibility.

5. Dynamic Learning from Music Preferences: The system is capable of learning user music preferences through deep learning models and that adaptation will be done dynamically. It will keep on refining its recommendations silently by constantly monitoring users skipped tracks, repeat plays, and ratings without user interaction.
6. Music Discovery for Emerging Artists: This platform can also help in bringing up of independent or emerging artists through feeding in music from off-mainstream sources. Utilizing emotion-based matching, one can effectively link niche music to the right audience which in turn, helps in preserving musical diversity.

V. CONCLUSION AND FUTURE SCOPE

The Chatbot Song Recommender System is a groundbreak- ing service that uses a talk interface for music recommendation based on mood detection via conversation. The emotions of the user are detected and analyzed through the collaboration of Natural Language Processing and emotional tone detection; thus, the interaction offered is more human-like and emotionally intelligent than those of the traditional recommendation engines that operate through user history or popularity trends alone. Successfully utilizing APIs like IBM Tone Analyzer and Last.fm has enabled the chatbot to interpret the mood of users dynamically and recommend songs that match the mood in real-time. Not only does this increase user interaction but also the whole process of discovering music becomes more enjoyable and personalized to one's emotions. The system, in general, is a perfect blend of cutting-edge machine learning, natural language processing, and user-oriented design, which allows people to engage with music services in a new way.

REFERENCES

1. M. Rohini, K. R. Naveena, G. Jothipriya, and S. Kameshwaran, "A comparative approach to predict Coronavirus using machine learning," in Proc. 2021 Int. Conf., IEEE, 2021.
2. S. Tharsha, J. Dilogera, B. Mohanashiyaam, S. Kirushan, K. B. N. H. P. Ravi, S. Swarnakantha, and A. B. Chathurika, "Machine learning-based prediction model for academic performance," in Proc. 2021 Int. Conf., IEEE, 2021.
3. V. Bhadana, A. S. Jalal, and P. Pathak, "A comparative study of machine learning models for COVID-19 prediction in India," in Proc. 2021 Int. Conf., IEEE, 2021. Fig. 3. Mood Analysed Playlists The Dashboard Functionality operated correctly. Performance Testing exhibited that the system had been able to hold its ground against heavy user traffic. Nonetheless, initial testing revealed some fields that needed improvement, especially in the areas of Real-Time Data Fetching and the Notification System which received a 'Fail' mark. However, the entire system logic, the sequential algorithm that moves from user input to tone analysis to Last.fm query, was dependable in terms of performance.
4. N. Darapaneni, P. Jain, R. Khattar, M. Chawla, R. Vaish, and A. R. Paduri, "Analysis and prediction of COVID-19 pandemic in India," in Proc. 2020 Int. Conf., IEEE, 2020.
5. L. Wang, H. Shen, K. Enfield, and K. Rheuban, "COVID-19 infection detection using machine learning," in Proc. 2021 Int. Conf., IEEE, 2021.

6. M. S. Alie, Y. Negesse, K. Kindie, D. Senay Merawi, et al., ”Machine learning algorithms for predicting COVID-19 mortality in Ethiopia,” BMC Public Health, vol. 24, art. 1728, June 2024.
7. A. Ramchandani, C. Fan, and A. Mostafavi, ”DeepCOVIDNet: An interpretable deep learning model for predictive surveillance of COVID- 19 using heterogeneous features,” in arXiv preprint, Jul. 2020.
8. A. S. Jokandan, H. Asgharnezhad, S. Shamsi Jokandan, A. Khosravi, P. Kebria, D. Nahavandi, S. Nahavandi, and D. Srinivasan, ”An uncertainty- aware transfer-learning-based framework for COVID-19 diagnosis,” in arXiv preprint, Jul. 2020.
9. I. D. Apostolopoulos and T. Bessiana, ”COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks,” in arXiv preprint, Mar. 2020.
10. Z. Hu, Q. Ge, S. Li, L. Jin, and M. Xiong, ”Artificial intelligence forecasting of COVID-19 in China,” in arXiv preprint, Feb. 2020. H. Gull, M. Saqib, S. Z. Iqbal, and S. Saeed, ”Improving learning experience of students by early prediction of student performance using machine learning,” in Proc. 2020 IEEE Int. Conf. Innovation inTechnology (INOCON), 2020.
11. R. Thomas and R. Gupta, ”Feature selection techniques and its importance in machine learning: A survey,” in Proc. IEEE Int. Students’ Conf. Electrical, Electronics and Computer Science (SCEECS), 2020.
12. T. R. Kumar et al., ”Students performance prediction using data mining techniques,” in Proc. 2019 Int. Conf. Intelligent Sustainable Systems (ICISS), 2019.
13. I. A. A. Amra and A. Y. A. Maghari, ”Students performance prediction using KNN and Na¨ıve Bayesian,” in Proc. 2017 8th Int. Conf. on Information Technology (ICIT), 2017.
14. M. Mohammadi, M. Dawodi, W. Tomohisa, and N. Ahmadi, ”Comparative study of supervised learning algorithms for student performance prediction,” in Proc. 2019 Int. Conf. on Artificial Intelligence in Information and Communication (ICAIIC), 2019.
15. S. A. Shah, ”Machine learning-based predictive analytics of student academic performance in STEM education,” in Proc. 2019 IEEE Global Eng. Educ. Conf., 2019.
16. H. Lalmuanawma, J. Hussain, and L. Chhakchhuak, ”Applications of machine learning and artificial intelligence for COVID-19 (SARS-CoV2) pandemic: A review,” Chaos Solitons Fractals, vol. 139, 2020.
17. C. Iwendi, A. K. Bashir, A. Peshkar, R. Sujatha, J. M. Chatterjee, S. Pasupuleti, et al., ”COVID-19 patient health prediction using boosted random forest algorithm,” Front. Public Health, vol. 8, 2020.
18. E. Cornelius, O. Akman, and D. Hrozencik, ”COVID-19 mortality prediction using machine learning-integrated random forest algorithm under varying patient frailty,” Mathematics, vol. 9, no. 17, 2021.
19. W. Booth, E. Abels, and P. McCaffrey, ”Development of a prognostic model for mortality in COVID-19 infection using machine learning,” Mod. Pathol., vol. 34, no. 3, pp. 522-531, 2021.