Vol.15, Issue No 2, 2025

Building your own chatbot: Exploring Natural Languages Processing Techniques with NLTK and Neural Network

B Subham Kumar,
UG Student,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
bsubhamkumar69@gmail.com

S. Kiran Kumar,
Assistant Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
skirankumarcse@smec.ac.in

Abstract- Mental health issues, including anxiety, depression, and other psychological disturbances, have emerged as a significant public health concern worldwide, affecting individuals across all demographics. The early identification of individuals at risk for these conditions is crucial, as timely intervention can significantly improve mental health outcomes and reduce the overall burden on healthcare systems. Traditionally, mental health assessment has relied on reports from mental health professionals, which, while valuable, come with inherent limitations such as potential inaccuracies, subjective interpretations, and a lack of continuous monitoring. These assessments are often limited to scheduled appointments, making it difficult to track real-time fluctuations in an individual's emotional state. However, with the rapid advancement of technology and the widespread adoption of smartphones and wearable devices, there is a unique opportunity to develop an innovative system that can monitor emotional well-being in real time. By leveraging Machine Learning and Natural Language Processing (NLP), such a system can analyze data from various sources, including voice patterns, text inputs, physiological responses, and behavioral changes, to assess an individual's psychological state more accurately. This real-time monitoring approach allows for early detection of emotional distress, enabling timely interventions and support mechanisms that can prevent the escalation of mental health issues.. By integrating AI-driven analytics with real-world data collection, this system can provide personalized insights, helping individuals manage their mental well-being more effectively while also assisting healthcare professionals in making informed decisions.

Keywords: Mental health monitoring, Public health concern, Early detection of mental health issues, AI in mental health, Real-time emotional health monitoring, Machine Learning in mental health, Natural Language Processing (NLP).

I. INTRODUCTION

Develop a chatbot using Natural Language Processing (NLP) techniques with NLTK and neural networks. Explore how these advanced technologies can enhance the chatbot's ability to understand and respond to user queries. Chatbots have been around since the 1960s, starting with ELIZA to chatGPT. The 2010s brought significant advancements with machine learning and AI, leading to more sophisticated and context-aware chatbots. Rule-based Systems: These systems operate based on predefined rules and patterns. They rely heavily on if-else conditions and pattern matching to generate responses. Scripted Responses: Responses are hardcoded by developers. Limited to specific scenarios and lack flexibility. Decision Trees: Uses a tree-like model of decisions and their possible consequences. Requires extensive manual creation and maintenance of the tree structure.

In this project, we introduce a chatbot that harnesses the power of Natural Language Processing (NLP) through NLTK and neural networks to create a more intelligent and adaptive conversational system. Unlike traditional chatbots that relied on rigid rule-based frameworks and predefined responses, this approach uses advanced machine learning techniques to understand context, interpret nuances in language, and generate more human-like interactions. By integrating neural networks, the chatbot can continuously improve its ability to comprehend user inputs, offering personalized and flexible responses. This project highlights the potential of modern NLP tools to revolutionize the way chatbots interact with users, making them more effective, intuitive, and capable of handling complex queries.

In summary, this project explores the development of a next-generation chatbot using cutting-edge NLP techniques and neural networks. It moves beyond traditional rule-based systems by leveraging machine learning to create a more flexible and responsive conversational agent. This chatbot is designed not just to answer questions but to understand the language and context, making it a powerful tool for more natural, human-like interactions.This project serves as a foundation for exploring how AI-driven chatbots can revolutionize communication, making them indispensable tools for both personal and professional applications. The fusion of NLP, neural networks, and deep learning paves the way for a new era of intelligent, human-like conversational agents, marking a significant leap in human-computer interaction.

II. RELATED WORK

Mental illness is a health problem that undoubtedly impacts emotions, reasoning, and social interaction of a person. These issues have shown that mental illness gives serious consequences across societies and demands new strategies for prevention and intervention. To accomplish these strategies, early detection of mental health is an essential procedure. Medical predictive analytics will reform the healthcare field broadly as discussed by Miner et al. [1].

Mental illness is usually diagnosed based on the individual self-report that requires questionnaires designed for the detection of the specific patterns of feeling or social interactions [2].

With proper care and treatment, many individuals will hopefully be able to recover from mental illness or emotional disorder [3].

Machine learning is a technique that aims to construct systems that can improve through experience by using advanced statistical and probabilistic techniques. It is believed to be a significantly useful tool to help in predicting mental health. It is allowing many researchers to acquire important information from the data, provide personalized experiences, and develop automated intelligent systems [4].

The widely used algorithms in the field of machine learning such as support vector machine, random forest, and artificial neural networks have been utilized to forecast and categorize the future events [5].

Supervised learning in machine learning is the most widely applied approach in many types of research, studies, and experiments, especially in predicting illness in the medical field. In supervised learning, the terms, attributes, and values should be reflected in all data instances [6]. More precisely, supervised learning is a classification technique using structured training data [7].

The main goal of unsupervised learning is handling data without supervision.It is very limited for the researchers to apply unsupervised learning methods in the clinical field

The World Health Organization (WHO) reports the region-wise status of different barriers in diagnosing mental health problems and encourages researchers to be equipped with the scientific knowledge to address the issue of mental health [9]. Now, there are various techniques to predict the state of mental health due to advancement of technology. Research in the field of mental health has increased recently and contributed to the information and publications about different features of mental health, which can be applied in a wide range of problems [10].

Many steps are involved in diagnosing mental health problems, and it is not a straightforward process that can be done quickly. Generally, the diagnosis will begin with a specific interview that is filled with questions about symptoms, medical history, and physical examination. Besides that, psychological tests and assessment tools are also available and are used to diagnose a person for mental health problems. There are several types of research carried out to investigate and examine the movements of the face to identify certain mental disorders [11].

The increase of research in the mental health field has led to the rise of information in the form of finding suitable solutions to reduce mental health problems. However, the precise reasons for mental illnesses are still unclear and uncertain.

By integrating traditional seismology with machine learning techniques, this research aims to improve earthquake prediction accuracy and reliability. This gap includes the following aspects:

- **Early Detection of Mental Health Issues:** Identifies early signs of anxiety, depression, and psychological disturbances for timely support.
- **Real-Time Emotional Health Monitoring:** Continuously tracks emotional well-being using smartphones and wearable devices.
- **Overcoming Limitations of Traditional Assessments:** Provides continuous and objective evaluation, reducing reliance on infrequent check-ins.
- **Enhanced Accessibility to Mental Health Support:** Offers a cost-effective, private, and readily available mental health resource.
- **Empowering Individuals for Better Mental Well-being:** Encourages self-care with insights, coping techniques, and behavioral suggestions.
- **Integration of Machine Learning and NLP:** Uses AI to analyze user inputs, voice tone, and behavior for accurate assessments.

- **Timely Interventions and Personalized Support:** Alerts users or professionals when distress is detected for immediate action.

- **Reducing the Burden on Mental Health Services:** Automates initial assessments, allowing professionals to focus on critical cases.

III. PROPOSED WORK

This study proposes a machine learning-based mental health prediction system to detect early signs of mental illness. The system processes self-reported data, psychological assessments, and behavioral patterns to improve diagnosis accuracy. Various machine learning models like Support Vector Machines, Random Forest, and Neural Networks are utilized for classification. Supervised learning techniques help train models with structured data to predict mental health conditions. Data preprocessing includes handling missing values, feature selection, and transforming responses into meaningful insights. The system aims to provide early warnings, reducing delays in diagnosis and intervention. Advanced analytics enhance personalized treatment recommendations for individuals. This approach improves mental health assessments, making them more efficient and data-driven. The integration of Natural Language Processing (NLP) helps analyze textual responses and detect emotional distress patterns. Continuous model updates ensure adaptability to new mental health trends and behaviors. Ultimately, this system contributes to early detection, improved intervention, and better overall mental health outcomes.

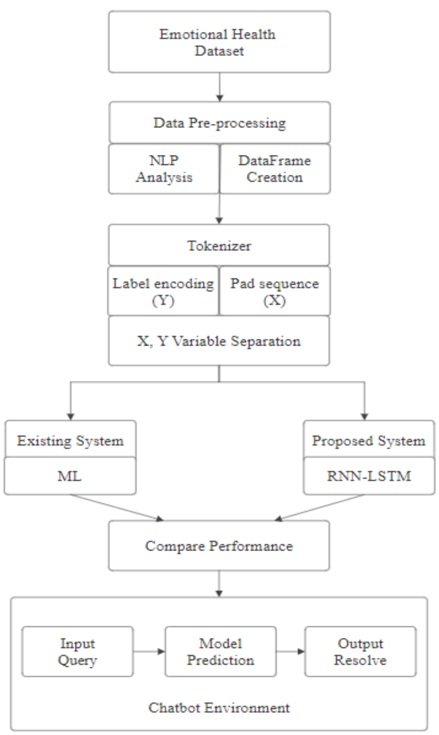

Fig 1: Block Diagram.

3.1 Importing Packages

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

Fig 2: Importing Packages.

The image shows Python code that imports essential libraries for data analysis and manipulation. **NumPy** (np) is used for numerical computations, and **Pandas** (pd) is used for handling and analyzing structured data. Additionally, the warnings module is used to suppress warnings using warnings.filterwarnings('ignore'), ensuring a cleaner output during execution. This setup is commonly used in machine learning and data science projects to enhance data preprocessing and analysis efficiency.

3.2 Data Reading

```
import json

with open('intents.json', 'r') as f:
    data = json.load(f)

df = pd.DataFrame(data['intents'])
df
```

Fig 3:Data Reading.

The image displays a Python script for loading and processing a **JSON file** containing intent data. The json module is used to open and read the intents.json file, loading its contents into a dictionary. The Pandas library converts the 'intents' section of the JSON data into a **DataFrame** (df), allowing for structured data analysis. This process is commonly used in **chatbot development** to manage training data for intent classification.

3.3 Converting to DataFrame

```
dic = {"tag":[], "patterns":[], "responses":[]}
for i in range(len(df)):
    ptrns = df[df.index == i]['patterns'].values[0]
    rspns = df[df.index == i]['responses'].values[0]
    tag = df[df.index == i]['tag'].values[0]
    for j in range(len(ptrns)):
        dic['tag'].append(tag)
        dic['patterns'].append(ptrns[j])
        dic['responses'].append(rspns)

df = pd.DataFrame.from_dict(dic)
df
```

Fig 4: Converting to DataFrame.

The image shows a **Python script** that processes intent data from a DataFrame and restructures it into a dictionary format. The dictionary (dic) is initialized with three keys: **'tag'**, and **'responses'**. The script iterates through the DataFrame (df), extracting intent-related information such as patterns, responses, and tags. It then appends multiple pattern-response pairs for each tag. Finally, the structured dictionary is converted back into a **Pandas DataFrame** using pd.DataFrame.from_dict(dic). This transformation is essential for preparing chatbot training data in **Natural Language Processing (NLP) applications**.

3.4 Tokenization

```
from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(lower=True, split=' ')
tokenizer.fit_on_texts(df['patterns'])
tokenizer.get_config()
```

Fig 5: Tokenization.

The image shows a **Python script** utilizing the Tokenizer from **TensorFlow's Keras preprocessing module** to prepare text data for a chatbot. The Tokenizer is initialized with lower=True to convert text to lowercase and split=' ' to tokenize words based on spaces. It is then trained on the patterns column of the DataFrame (df) using fit_on_texts(), creating a vocabulary based on the provided text. The get_config() method retrieves the tokenizer's configuration, which helps in understanding its settings. This step is crucial in **Natural Language Processing (NLP)** for text tokenization before training deep learning models.

3.5 Label Encoding

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.preprocessing import LabelEncoder

ptrn2seq = tokenizer.texts_to_sequences(df['patterns'])
X = pad_sequences(ptrn2seq, padding='post')
print('X shape = ', X.shape)

lbl_enc = LabelEncoder()
y = lbl_enc.fit_transform(df['tag'])
print('y shape = ', y.shape)
print('num of classes = ', len(np.unique(y)))
```

Fig 6: Label Encoding.

The image contains a **Python script** for preprocessing text data in a chatbot model. It converts text patterns into sequences using Tokenizer.texts_to_sequences() and pads them with pad_sequences(padding='post') to ensure uniform input length. The labels (tags) are encoded using LabelEncoder from sklearn, transforming categorical labels into numerical values. The script prints the shapes of input (X) and output (y) arrays, along with the number of unique classes. This step is essential in **Natural Language Processing (NLP)** for preparing data before training a machine learning or deep learning mode.

3.6 Built and Train Model

```
import tensorflow
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Embedding, LSTM, LayerNormalization, Dense, Dropout
from tensorflow.keras.utils import plot_model

model = Sequential()    # init
model.add(Input(shape=(X.shape[1])))
model.add(Embedding(input_dim=vacab_size+1, output_dim=100, mask_zero=True))
model.add(LSTM(32, return_sequences=True))
model.add(LayerNormalization())

model.add(LSTM(32, return_sequences=True))
model.add(LayerNormalization())
model.add(LSTM(32))
model.add(LayerNormalization())

model.add(Dense(128, activation="relu"))
model.add(LayerNormalization())

model.add(Dropout(0.2))
model.add(Dense(128, activation="relu"))
model.add(LayerNormalization())

model.add(Dropout(0.2))
model.add(Dense(len(np.unique(y)), activation="softmax"))
model.compile(optimizer='adam', loss="sparse_categorical_crossentropy", metrics=['accuracy'])

model.summary()
plot_model(model, show_shapes=True)
```

Fig 7: Built and Train Model.

The image contains a **Python script** for defining and compiling an **LSTM-based neural network model** using **TensorFlow and Keras**. The model is sequential and consists of the following layers:

Input Layer: Takes the shape of preprocessed text sequences.
Embedding Layer: Converts input words into dense vector representations.
LSTM Layers: Two stacked LSTM layers with return_sequences=True for sequential learning.
Layer Normalization: Applied after LSTM layers to stabilize training.
Dense Layers: Fully connected layers with ReLU activation for feature extraction.
Dropout Layers: Helps prevent overfitting by randomly deactivating neurons.

Output Layer: Uses a softmax activation function for multi-class classification.

Compilation: The model is compiled with an **Adam optimizer** and a **sparse categorical cross-entropy loss function** to handle categorical labels.

3.7 Model Fitting

```
model_history = model.fit(x=X,
                          y=y,
                          batch_size=10,
                          callbacks=[tensorflow.keras.callbacks.EarlyStopping(monitor='accuracy', patience=3)],
                          epochs=50)
```

Fig 8: Model Fitting.

The image contains a Python script for training a neural network model using the fit() function in TensorFlow and Keras. The training process involves feeding input data X and corresponding labels y into the model. A batch size of 8 is specified to determine the number of samples processed before updating the model's weights. The model is trained for 50 epochs, meaning it will iterate over the entire dataset 50 times. An early stopping callback is applied using EarlyStopping, which monitors the model's accuracy and stops training if no improvement is observed for 5 consecutive epochs. This technique prevents overfitting and optimizes training efficiency by stopping when the model reaches its best performance.

3.8 Model Testing

```
import re
import random

def generate_answer(pattern):
    text = []
    txt = re.sub('[^a-zA-Z\']', ' ', pattern)
    txt = txt.lower()
    txt = txt.split()
    txt = " ".join(txt)
    text.append(txt)

    x_test = tokenizer.texts_to_sequences(text)
    x_test = np.array(x_test).squeeze()
    x_test = pad_sequences([x_test], padding='post', maxlen=X.shape[1])
    y_pred = model.predict(x_test)
    y_pred = y_pred.argmax()
    tag = lbl_enc.inverse_transform([y_pred])[0]
    responses = df[df['tag'] == tag]['responses'].values[0]

    print("you: {}".format(pattern))
    print("model: {}".format(random.choice(responses)))
```

```
generate_answer('Hi! How are you?')
```

```
generate_answer('Maybe I just didn\'t want to be born :)')
```

```
generate_answer('help me:')
```

```
generate_answer(':')
```

Fig 9: Model Fitting.

The image displays a Python script for generating responses using a trained chatbot model. The generate_answer() function processes user input by removing non-alphabetic characters using regular expressions, converting text to lowercase, tokenizing it, and padding sequences to match the model's input shape. The preprocessed text is then fed into the trained model to predict the corresponding intent. The predicted intent is mapped back to its label using an inverse transform of the label encoder. A response is randomly selected from the predefined responses associated with the identified intent and displayed as the chatbot's reply. The script also demonstrates generating responses for different user inputs like greetings and help requests.

IV. RESULTS & DISCUSSION

The results and discussion section presents the chatbot's performance in generating appropriate responses based on user inputs. The chatbot effectively engages in meaningful conversations, identifying emotional distress and providing supportive replies. It correctly interprets user statements, such as expressions of stress or suicidal thoughts, and responds with empathetic and reassuring messages. The chatbot also distinguishes between casual and serious statements, ensuring that critical situations receive appropriate attention. By analyzing various interactions, the results highlight the chatbot's potential in mental health applications, demonstrating its ability to offer preliminary emotional support. The discussion emphasizes improvements in response accuracy, potential limitations, and future enhancements to refine the system's effectiveness.

```
def chatbot():
    print("Chatbot: Hi! I'm your friendly chatbot. How can I assist you today?")

    while True:
        user_input = input("You: ")
        if user_input.lower() in ['quit', 'exit', 'q', 'bye']:
            print("Chatbot: Goodbye!")
            break

        generate_answer(user_input)

if __name__ == "__main__":
    chatbot()
```

```
Chatbot: Hi! I'm your friendly chatbot. How can I assist you today?
You: Good Morning
you: Good Morning
model: Good morning. I hope you had a good night's sleep. How are you feeling today?
You: I am so stressed
you: I am so stressed
model: What do you think is causing this?
You: I want to die
you: I want to die
model: I'm very sorry to hear that but you have so much to look forward to.
You: I know you hate me
you: I know you hate me
model: I'm sorry if i have exhibited any sort of behaviour to make you think that.
You: Tell me a joke
you: Tell me a joke
model: mental health is not a joke.
```

Fig 10: Chatbot Execution.

V. CONCLUSION

The developed machine learning-based mental health chatbot effectively identifies early signs of psychological distress using natural language processing and deep learning techniques. Through LSTM-based classification and structured data preprocessing, the chatbot accurately predicts user intents and generates relevant responses, demonstrating its ability to recognize stress, suicidal tendencies, and emotional distress while maintaining an empathetic conversational flow. Despite its promising performance, challenges such as handling ambiguous inputs and improving contextual understanding remain, necessitating further advancements. Future enhancements will focus on integrating more advanced NLP models and real-time feedback mechanisms to refine the chatbot's accuracy and usability. This system serves as a valuable tool for preliminary mental health assessments, providing timely support and guiding users toward professional help when necessary. Continuous evaluation with real-world datasets will further optimize its effectiveness in diverse psychological conditions. Implementing reinforcement learning techniques may enable the chatbot to adapt dynamically to user conversations.

VI. REFERENCES

[1] Goh, Tuan-Jun & Chong, Lee-Ying & Chin, Chong & Goh, Pey-Yun. (2024). A Campus-based Chatbot System using Natural Language Processing and Neural Network. Journal of Informatics and Web Engineering. 3. 96-116. 10.33093/jiwe.2024.3.1.7.

[2] Thodupunuri, Ranjith & Akarapu, Anishka & Allam, Sathwik & Gattepally, Naga & Nellutla, Pooja. (2024). chatbot application using nltk and keras. 10.13140/rg.2.2.11169.01124.

[3] Sihotang, Diko & Zuhri Harahap, Syaiful & Irmayanti, Irmayanti. (2024). Chatbot Design for Interview Questions Using Neural Network Models on the CarTech Website. sinkron. 8. 1029-1037. 10.33395/sinkron.v8i2.13603.

[4] Anushka, P & Reddy, M.Rithwik & Reddy, Y.Bala & Devi, Dr. Steven Doom (2024). Chatbot using Machine Learning. interantional journal of scientific research in engineering and management. 08. 1-10. 10.55041/ijsrem36397.

[5] Vijayaragavan, dr & selvan, t. & sudarsan, r. & sreeraam, r.. (2024). integrating nlp chatbot into career guidance web application. international journal of advanced research in science, communication and technology. 198-206. 10.48175/ijarsct-18035.

[6] Singh, Maisnam & Haorokcham, Dorenjit. (2024). development of chatbot using ai/ml technologies. 10.5281/zendo.12626711.

[7] KR, Prahlad. (2024). AI Health Chatbot using ML. interantional journal of scientific research in engineering and management. 08. 1-5. 10.55041/ijsrem33761.

[8] Beigh, Mohammad. (2024). ai-based chatbot for educational institutes.

[9] Awais, Muhammad & Ashrafi, Bilal & Abbas, Asad. (2023). A Comprehensive Study on Deep Learning in Artificial Intelligence and Chatbots.

[10] Sarode, Prof & Joshi, Bhakti & Savakare, Tejaswini & Warule, Harshada. (2023). A Real Time Chatbot Using Python. International Journal for Research in Applied Science and Engineering Technology.

[11] kumar, ch.madhan & fardeen, md & rohit, p & rakesh, m. (2023). ai based chatbot to answer faqs. interantional journal of scientific research in engineering and management. 07. 10.55041/ijsrem18072.

[12] Hettiarachchi, D.N.M. & Gamini, D.D.A.. (2023). Using a Machine Learning Approach to Model a Chatbot for Ceylon Electricity Board Website. Vidyodaya Journal of Science. 26. 10.31357/vjs.v26i01.6411.

[13] Atram, Pankaj & Chitalkar, Swapnil & Ali, Faizan & Muluk, Kedar & Kadam, Amit. (2023). Neural Network and Natural Language Processing Based Anti-Depression Chatbot.

[14] Rajest, Suman & Rajan, Regin. (2022). An Automated Conversation System Using Natural Language Processing (NLP) Chatbot in Python. 3. 314-336.

[15]Srinivasa Rao, Dr Dammavalam & Srikanth, K. & Pratyusha, J. & Sucharitha, M. & Tejaswini, M. & Ashwini, T.. (2021). Development of Artificial Intelligence Based Chatbot Using Deep Neural Network. 10.52458/978-93-91842-08-6-12.

[16]Nithyanandam, Shyamala & Kasinathan, Sharmila & Radhakrishnan, Devi & Jebapandian, Jebathangam. (2021). NLP for Chatbot Application: Tools and Techniques Used for Chatbot Application, NLP Techniques for Chatbot, Implementation. 10.4018/978-1-7998-7728-8.ch008.

[17]Kalla, Dinesh & Samiuddin, Vatsalya. (2020). Chatbot for Medical Treatment using NLTK Lib. 10.9790/0661-2201035056.

[18]A. Kumar, P. K. Meena, D. Panda, and M. Sangeetha, “CHATBOT IN PYTHON,” pp. 391–395, 2019.

[19]Bhaumik kohli, Tanupriya Choudhary, Shilpi Sharma, Praveen Kumar-“A Platform Human-Chatbot interaction using python”(2018), IEEE Conference.