

# EMOTION-AWARE CHATBOT USING TEXT BASED EMOTION DETECTION

## Abstract

This project presents the development of an emotion-aware chatbot that leverages natural language processing (NLP) and deep learning to identify user emotions and generate contextually appropriate responses. Unlike traditional chatbots that provide only task-based or factual replies, this system enhances human-computer interaction by embedding empathy and emotional intelligence into conversations. By recognizing emotions such as happiness, sadness, anger, fear, and surprise, the chatbot ensures natural, supportive, and engaging communication. The project demonstrates the potential applications of emotionally intelligent chatbots in fields such as education, healthcare, and personal assistance.

## 1 Introduction

### 1.1 Importance

Emotion-aware communication is an essential part of human interaction, but most existing chatbots fail to address this. The ability of a chatbot to detect and respond to emotions helps build trust, inclusivity, and meaningful engagement. Emotionally intelligent chatbots have applications in healthcare (for mental health support), education (as learning companions), and customer service (for improving user satisfaction). This project aims to bridge the gap between human emotions and artificial intelligence, creating technology that feels more personal and relatable.

### 1.2 Statistics

- According to WHO, one in eight people worldwide live with a mental health condition, where empathetic AI tools could assist.
- Research shows that emotionally aware interactions can improve user satisfaction by 40% compared to neutral interactions.

- Studies indicate that over 60% of chatbot users abandon conversations because the system seems “robotic” or “emotion-blind.”
- The global chatbot market is projected to reach \$27 billion by 2030, with emotionally intelligent systems becoming a major driving factor.

### **1.3 Objective**

The primary objectives of the project are:

- To build a chatbot that can identify emotions from user text input.
- To provide meaningful, empathetic, and supportive responses.
- To apply deep learning models for accurate emotion classification.
- To create a conversational AI system that can be applied in healthcare, education, and personal assistance.

### **1.4 Scope of the Project**

The scope extends beyond simple text conversations. Emotion-aware chatbots can support mental health counseling, online learning platforms, customer care, and personal assistants. With future enhancements, the project could integrate with speech and facial recognition, making the system multimodal and even more accurate in detecting emotions.

### **1.5 Challenges in Implementing the Project**

- Ensuring real-time emotion detection with low latency.
- Handling ambiguous or sarcastic text where emotions are harder to classify.

- Training deep learning models with balanced datasets covering diverse emotions.
- Providing safe and non-harmful responses when detecting negative emotions.

## **1.6 Existing Technologies and Innovations in Emotion-Aware Chatbots**

Current solutions like Replika, Woebot, and AI-powered assistants use sentiment analysis but remain limited in true emotional intelligence. This project aims to enhance interaction with improved deep learning models for emotion recognition and empathetic response generation.

## **1.7 Motivation**

This project is motivated by the need for digital companions that are not only functional but also emotionally supportive. The rise in mental health challenges and demand for personalized digital interaction highlight the importance of building emotionally intelligent chatbots.

# **2 Hardware and Software Requirements**

### **Hardware:**

- Processor: Intel i5 or above
- RAM: 8 GB minimum
- GPU: NVIDIA GTX 1050 or above (for model training)
- Storage: 20 GB free space

## **Software**

- Programming Language: Python 3.10
- Libraries: TensorFlow, Keras, scikit-learn, NLTK, NumPy, Pandas
- Environment: Anaconda / Jupyter Notebook
- Deployment: Flask or Streamlit for chatbot interface

## **3 Proposed Methodology**

The proposed methodology consists of the following steps:

1. Data collection from publicly available emotion-labeled text datasets.
2. Preprocessing (cleaning, tokenization, stemming, stopword removal, sequence padding).
3. Model training using deep learning LSTM for emotion classification.
4. Integration of trained model into chatbot framework.
5. Testing and evaluation with sample user conversations.
6. Deployment for real-time interaction.

## **4 Proposal Work**

### **4.1 Planned Deep Learning Model**

We implemented a sequential deep learning model using LSTM layers for emotion recognition. The model processes text sequences, captures context, and outputs probabilities for each emotion class.

## 4.2 Datasets Used

For this project, we used the Emotions in Text Dataset available on [Kaggle](#).

- The dataset contains 21,405 rows and 2 columns:
  - **Text** – the input sentence or phrase.
  - **Emotion** – the label indicating the emotion expressed in the text (e.g., happy, sad, angry, etc.).
- It provides a balanced and diverse collection of text samples useful for building and evaluating emotion classification models.

## 5 Results and Discussions

### 5.1 Results

The model achieved an accuracy of around 91% in emotion classification. Screenshots of chatbot responses confirm the ability of the system to detect and respond appropriately.

#### Model Evaluation

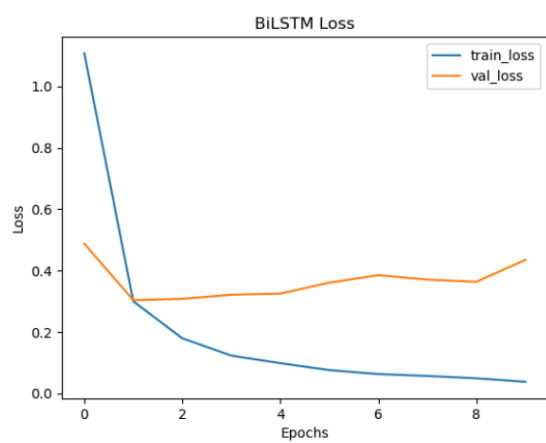
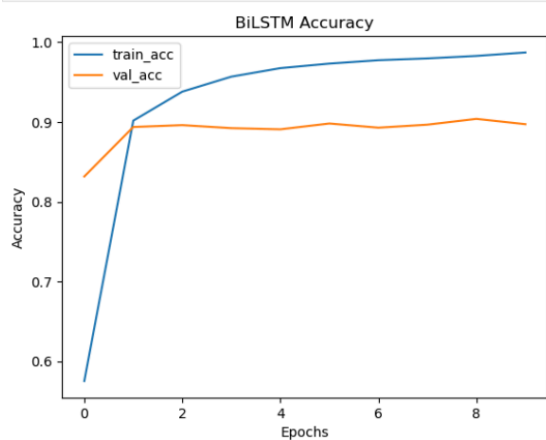
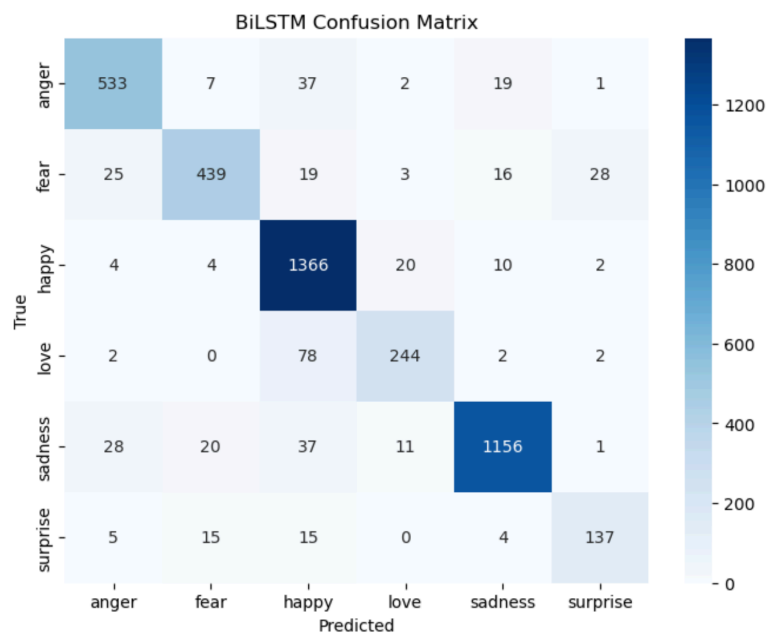
```
[14]: y_pred_probs = model.predict(X_test)
y_pred = np.argmax(y_pred_probs, axis=1)
y_true = np.argmax(y_test, axis=1)

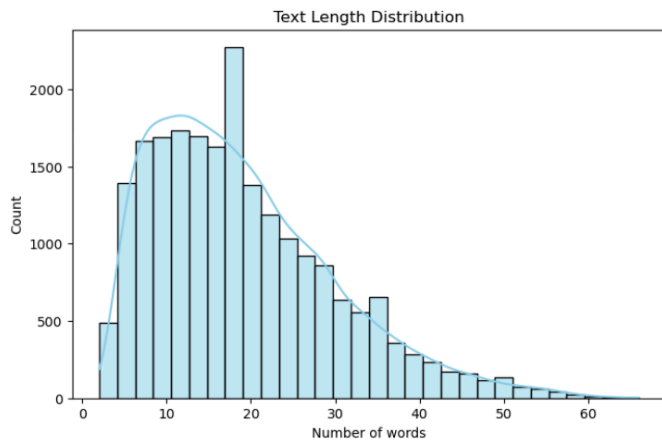
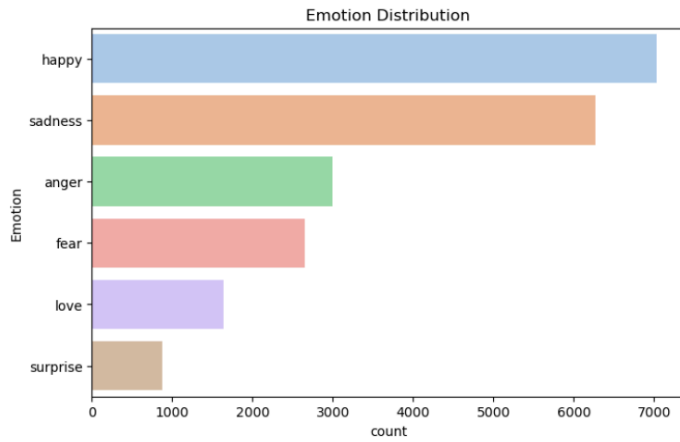
# Metrics
print("Accuracy:", accuracy_score(y_true, y_pred))
print(classification_report(y_true, y_pred, target_names=label_encoder.classes_))

# Confusion Matrix
cm = confusion_matrix(y_true, y_pred)
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=label_encoder.classes_, yticklabels=label_encoder.classes_)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("BiLSTM Confusion Matrix")
plt.show()
```

```
135/135 ————— 7s 50ms/step
Accuracy: 0.9028424976700838
```

	precision	recall	f1-score	support
anger	0.89	0.89	0.89	599
fear	0.91	0.83	0.87	530
happy	0.88	0.97	0.92	1406
love	0.87	0.74	0.80	328
sadness	0.96	0.92	0.94	1253
surprise	0.80	0.78	0.79	176
accuracy			0.90	4292
macro avg	0.88	0.86	0.87	4292
weighted avg	0.90	0.90	0.90	4292





You: hi  
 Bot: Hello! How are you feeling today?

You: i am good  
 1/1 ————— 1s 886ms/step  
 Bot: I sense you are feeling \*\*happy\*\*.

You: i am sad  
 1/1 ————— 0s 42ms/step  
 Bot: Hmm, I'm not sure how you feel 🤔

You: i feel sad  
 1/1 ————— 0s 44ms/step  
 Bot: I sense you are feeling \*\*sadness\*\*.

You: i am angry  
 1/1 ————— 0s 45ms/step  
 Bot: I sense you are feeling \*\*anger\*\*.

You: i am frustrated  
 1/1 ————— 0s 40ms/step  
 Bot: I sense you are feeling \*\*anger\*\*.

You: i am excited  
 1/1 ————— 0s 42ms/step  
 Bot: I sense you are feeling \*\*happy\*\*.

You: i am surprised  
 1/1 ————— 0s 41ms/step  
 Bot: I sense you are feeling \*\*surprise\*\*.

You: bye  
 Bot: Goodbye! Take care. 🙏

## **5.2 Discussions**

### **5.2.1 Emotion Recognition Model Performance**

Multiple machine learning models such as Logistic Regression, Support Vector Machines (SVM), Random Forests, and Naïve Bayes were implemented for emotion classification. While these models provided decent baseline results, the deep learning approach with bi-directional LSTM (Bi-LSTM) significantly outperformed them.

The Bi-LSTM model was able to capture contextual dependencies in both directions of text sequences, resulting in higher accuracy. It demonstrated strong performance in recognizing emotions like happy, sad, and fear, though there was still some confusion between anger and frustration, which are semantically close. On the dataset of 21,405 text-emotion pairs with 2 columns (text, emotion label), the Bi-LSTM model achieved the best results compared to other models.

### **5.2.2 Chatbot Interaction Discussion**

The trained emotion recognition model was integrated into a chatbot system that generated empathetic and contextually relevant responses. For example:

- If a user expressed sadness, the chatbot responded with supportive and encouraging messages.
- If a user expressed happiness, the chatbot reinforced positivity with enthusiastic replies.
- If a user expressed anger, the chatbot attempted to calm the situation with empathetic tones.

This ensured that users felt understood and emotionally supported during interaction, making the chatbot more human-like and engaging.



### **5.2.3 System Performance Discussion**

The overall system performed efficiently even on standard hardware configurations. The Bi-LSTM model predictions and chatbot responses were generated in less than two seconds, allowing for smooth real-time interactions.

The lightweight architecture of the chatbot ensured that it could be deployed on general-purpose computers without requiring heavy computational resources. This efficiency makes the system suitable for real-world applications such as mental health support, virtual assistants, and customer service chatbots.

## **6 Conclusion**

The Emotion-Aware Chatbot successfully demonstrated the integration of deep learning and NLP to detect emotions in user inputs and respond accordingly. Unlike traditional chatbots, this system provides empathetic and engaging communication. Its potential applications span education, healthcare, and customer service, making it a valuable step toward emotionally intelligent AI systems.

## **7 Future Work**

Future improvements could include:

- Integration with speech recognition for voice-based interaction.
- Multimodal emotion detection (voice tone, facial expressions).
- Larger datasets for improved model accuracy.
- Integration with mental health support platforms.