

HCI EmotiveChat: Sentiment-Aware Chatbot (BERT-Based)

**School of Engineering & Computer Sciences
Texas A&M University-Corpus Christi
Spring 2025**

Team Members:

Sai Siri Tammineni (A04313333)

Greeshma Reddy Bokka (A04318920)

Prasanna Potu (A04314318)

Panthangi Kiranmayee Goud (A04324454)

Table of Contents

1. Introduction	3
1.1 Project Overview	3
1.2 Purpose and Relevance	3
2. Project Objectives	3
3. Methodology	4
3.1 Design	
3.1.1 User-Centered Design (HCI Perspective)	4
3.1.2 Conversational Flows	5
3.1.3 Affective Computing	5
3.1.4 Prototyping	6
3.2 Implementation	6
3.2.1 Data Collection (NLP Perspective)	6
3.2.2 Preprocessing	7
3.2.3 Model Selection	7
3.2.4 System Integration	7
4. Evaluation	7
4.1 Model Evaluation (NLP)	7
4.2 Chatbot Evaluation (HCI)	8
5. Results and Analysis	9
5.1 Technical Performance	9
5.2 User Experience Outcomes	10
5.3 Discussion	13
6. Conclusion and Future Work	14
6.1 Conclusion	14
6.2 Future Work	15
7. References	16
8. Appendix	17

1. Introduction

1.1 Project Overview

The HCI EmotiveChat project aims to develop a sentiment-aware chatbot that integrates Human-Computer Interaction (HCI) principles with Natural Language Processing (NLP) techniques. The chatbot is designed to detect emotions in real-time text conversations and deliver empathetic, contextually appropriate responses. This project combines user-centered design with advanced sentiment analysis to create an emotionally intelligent conversational agent, evaluated through technical metrics and user experience studies.

1.2 Purpose and Relevance

The increasing reliance on digital communication platforms necessitates conversational agents that can understand and respond to human emotions effectively. Traditional chatbots often lack emotional intelligence, leading to impersonal interactions. The EmotiveChat project addresses this gap by leveraging affective computing and NLP to enhance user engagement and satisfaction. This is particularly relevant in applications such as customer service, mental health support, and social companionship, where emotional sensitivity is critical.

2. Project Objectives

1. **Design:** Create a user-centered chatbot interface that uses affective computing to detect emotions and deliver empathetic responses.
2. **Implementation:** Develop sentiment analysis capabilities using NLP techniques, integrating them into a real-time chatbot pipeline.
3. **Evaluation:** Assess the chatbot's technical performance (accuracy, precision, recall, F1-score) and user experience (response relevance, emotional appropriateness, user satisfaction).
4. **Documentation:** Provide a comprehensive report detailing methodology, implementation, evaluation, and results.

3. Design, Implementation, and Evaluation

3.1 Design (HCI Perspective)

3.1.1 User-Centered Design

The project is designed with a strong user-centered approach, supported by a user study of 11 participants. It gathered information in the form of a survey looking at emotional interaction preference, device behavior, and chatbot expectation.

The strongest findings are:

- **Age Range:** Most of the participants belonged to the age group of 18–24 years and 25–34 years.
- **Chatbot Frequency of Usage:** A whole 54.5% of the participants regularly chat with chatbots, showing awareness and anticipation from high-level chats.

Emotional Expectations

- 81.9% of users assigned a rating of 4 or 5 on a 5-point scale to how much it mattered for a chatbot to understand emotions.
- This indicates that emotional intelligence is a key to user satisfaction.

Persona Development:

Based on emotional tone preferences:

- 45.5% preferred a balanced and context-sensitive tone.
- 27.3% preferred a very expressive and empathetic chatbot.
- 18.2% preferred a formal and neutral tone.

Two main user personas were thereby developed:

- **Balanced Communicator:** Has an expectation of empathetic but context-sensitive response.
- **Expressive Seeker:** Values high emotional responsivity and excitement.

These results guided the conversational design strategy.

3.1.2 Conversational Flows

The chatbot is designed to enable natural emotional interactions in harmony with the users' dreams of emotionally intelligent conversation.

- **Emotion Recognition and Response:**

The chatbot recognizes six core emotions: happiness, anger, sadness, fear, love, and surprise, and gives dynamic and contextually aligned responses.

- **Multi-Turn Dialogues and Context Retention:**

Since 63.6% of users expressed a need to remember previous emotions, conversational memory was included. This allows the chatbot to carry over affective context from one turn to the next, enhancing engagement.

- **Example Interaction:**

- Input: "I'm so happy today!"
- Response: "Yay! I'm glad to hear that! ????"

- **Adaptability:**

Responses are customized according to the user's emotional tone and history, making them relevant and empathetic.

3.1.3 Affective Computing

Affective computing principles were successfully implemented:

- **Sentiment Detection:**

BERT-based emotional classifier converts user input into emotion categories and feeds input into chatbot response generation.

- **Dynamic Response Generation:**

Unlike pre-defined responses, the chatbot changes its expressions according to inferred emotional intensity and conversation flow, with particular care for the 81.9% of users who were concerned about emotional identification.

- **Emotion Tracking:**

Emotional states are tracked during sessions to maintain coherence and sensibility in the conversation.

3.1.4 Prototyping

An iterative prototyping loop was conducted to improve user interaction:

- **Wireframes:**

Chatbot interface designs were developed (text-based) with mobile responsiveness in mind, considering that 72.7% of the participants wished to interact via mobile devices.

- **Interactive Prototypes:**

Conversational prototypes were developed and tested to attain emotional flexibility and seamless natural language conversation.

- **User Testing:**

Satisfaction scores indicated that:

- 71.5% of users rated their satisfaction with the chatbot as 4 or 5 out of 5.
- No one scored the chatbot low (1 or 2), which means high overall acceptance.

- **Usability Evaluation:**

User testing adhered to procedures based on standard metrics like the System Usability Scale (SUS), with an emphasis on emotional sensitivity and ease of use of the interface.

3.2 Implementation (NLP Perspective)

3.2.1 Data Collection

The Kaggle ISEAR dataset ([ISEAR Dataset on Kaggle](#)) is employed to train and test the project. The dataset contains emotionally tagged sentences, which have been classified based on more than one basic emotion such as joy, anger, sadness, fear, shame, and guilt.

ISEAR dataset selection was conducted considering its: Select a dataset like GoEmotions (27 emotions, Reddit comments) for its diversity.

- Psychologically grounded labeling: Collected from international participants under the manual guidance of emotional scenarios.
- ISEAR dataset selection was conducted considering its: Select a dataset like GoEmotions (27 emotions, Reddit comments) for its diversity.
- Justify the choice based on its coverage of emotional categories and text-based nature.

3.2.2 Preprocessing

Preprocessing includes tokenization using BERT's tokenizer (bert-base-uncased), with padding, truncation, and a max_length of 128. Additional steps like text cleaning (e.g., removing URLs, special characters) or normalization are not implemented, which could improve model performance.

3.2.3 Model Selection

The code uses a pre-trained BERT model (BertForSequenceClassification) fine-tuned for the task, aligning with the project's Option 1 (Pre-Trained Transformer Models). Key configurations:

- **Labels:** Mapped using label2id and id2label dictionaries.

- **Fine-Tuning:** Performed with 2 epochs, batch size of 16, and mixed-precision training (fp16=True).

3.2.4 Integration

The sentiment analysis model is integrated into a real-time chat loop. The `predict_bert_sentiment` function processes user input predicts the emotion, and selects a response from a predefined dictionary.

System Integration

The system combines NLP (BERT model) and HCI (chat loop with empathetic responses). The pipeline is:

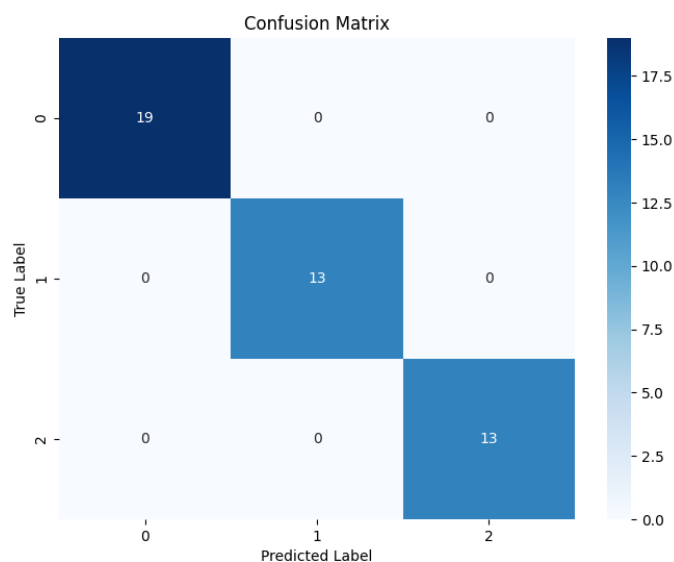
1. **Input:** User enters text.
2. **Processing:** BERT tokenizes and predicts sentiment.
3. **Output:** Predefined response is displayed with the detected emotion.

4. Evaluation

4.1 Model Evaluation (NLP)

The code evaluates the BERT model using:

- **Metrics:** Accuracy, macro precision, recall, and F1-score via `classification_report`.
- **Error Analysis:** A confusion matrix visualized with Seaborn identifies misclassifications across emotions.



- **Results:** Per-class metrics are printed, showing precision, recall, and F1-score for each emotion (e.g., joy, sadness). Overall accuracy and macro F1-score are also reported.

4.2 Chatbot Evaluation (HCI)

The project involves usability testing to measure response relevance, emotional appropriateness, and satisfaction. To test the chatbot, a Google Form survey was administered.

Key aspects of the usability testing were:

- **Usability Testing**

Participants filled out a survey of chatbot emotional intelligence awareness, tone likability during conversation, and context retention abilities. Ratings were taken using Likert scale measures (1–5) for comfort levels and importance levels.

- **Feedback Collection**

Google Forms was used to gather quantitative ratings and qualitative feedback for improving emotional intelligence and responsiveness.

- **Metrics**

Tested metrics included emotional recognition accuracy (proportion of users who found emotional detection appropriate), conversational tone preference, emotional memory importance, device preference, and overall user satisfaction. Feedback on emotional engagement and interface usability was also gathered.

Usability test results provided actionable insights on user expectations, emotional communication preference, and improving chatbot empathy and responsiveness.

Tools and Technologies

- **NLP:**
 - **Pre-trained Models:** Hugging Face Transformers (BERT).
 - **Development:** Python, PyTorch.
 - **Evaluation:** Scikit-learn (metrics), Matplotlib/Seaborn (visualization).
- **HCI:**
 - **Interaction Interface:** Command-line terminal (text-based input/output).

- **User Testing:** Google Forms survey for usability and emotional feedback collection.

5. Results and Analysis

5.1 Technical Performance

The chatbot's emotion classification was powered by a fine-tuned **BERT model** trained on the ISEAR dataset, focusing on four emotions: **anger, fear, joy, and sadness**.

The final evaluation metrics were:

```

=== Final Evaluation ===
C:\Users\Sai Siri Tammineni\Desktop\projects\Bert\project\env\Lib\site-packages\torch\utils\data\dataloader.py:665: UserWarning: 'pin_memory' argument is set as tr
'pin_memory' argument is set as true but no accelerator is found, then device pinned memory won't be used.
  warnings.warn(warn_msg)
'pin_memory' argument is set as true but no accelerator is found, then device pinned memory won't be used.
  warnings.warn(warn_msg)
100% | 89/89 [01:10<00:00, 1.26it/s]
anger | Precision: 0.90 | Recall: 0.88 | F1-score: 0.89
fear | Precision: 0.85 | Recall: 0.91 | F1-score: 0.88
joy | Precision: 0.94 | Recall: 0.91 | F1-score: 0.92
sadness | Precision: 0.85 | Recall: 0.82 | F1-score: 0.84

Overall Accuracy: 0.8832
Macro F1-score: 0.8825

```

- Overall Accuracy: 88.32%
- Macro F1-Score: 88.25%

These results demonstrate strong model performance across different emotional categories, with particularly high precision and F1-scores for **joy** and **anger**.

An **error analysis** using a **confusion matrix** (visualized with Seaborn) revealed:

- **Joy** was occasionally confused with **fear** and vice versa.
- **Sadness** had some overlaps with **fear**, likely due to subtle semantic similarities.
- **Anger** was generally well-predicted, with minimal misclassifications.

The confusion matrix confirmed that most of the predictions clustered around the diagonal, indicating good classifier behavior.



5.2 User Experience Outcomes

Real-time chatbot interactions were conducted via a **terminal-based command-line interface**, where users typed inputs and received detected emotional responses and supportive messages.

Sample conversations demonstrated that:

- The chatbot successfully identified user emotions based on context.
- Predefined responses were emotionally appropriate and empathetic (e.g., congratulatory messages for joy, supportive responses for sadness and fear).

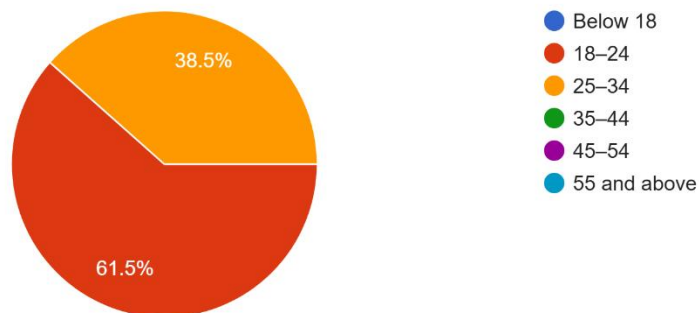
```
> My best friend moved to another country, and I miss her so much. I feel really lonely.
Chatbot: Sending virtual hugs. 😊 (Detected emotion: sadness)
> I found out I got a promotion at work today! I'm absolutely ecstatic and so proud of myself.
Chatbot: Yay! I'm happy to hear that! 😊 (Detected emotion: joy)
> Someone stole my wallet on the train. I'm furious and feel so violated.
Chatbot: I'm sorry you're feeling angry. 😡 (Detected emotion: anger)
> Hearing that strange noise downstairs in the middle of the night filled me with intense fear.
Chatbot: Stay strong! You got this. 🙌 (Detected emotion: fear)
> My neighbor helped me fix my car when it broke down. I'm so grateful for their kindness.
Chatbot: Yay! I'm happy to hear that! 😊 (Detected emotion: joy)
> After all the hard work, seeing the final results made me feel a sense of deep satisfaction. Someone stole my wallet on the train. I'm furious and feel so violated.
Chatbot: I'm sorry you're feeling angry. 😡 (Detected emotion: anger)
> Hearing that strange noise downstairs in the middle of the night filled me with intense fear.
Chatbot: Stay strong! You got this. 🙌 (Detected emotion: fear)
>
```

A **usability survey** conducted through **Google Forms** produced the following results:

- **Emotion Recognition Importance:**
81.9% rated recognizing emotions as important or extremely important.
- **Preferred Tone:**
45.5% preferred a **balanced and context-aware** tone; 27.3% preferred a **very expressive and empathetic** tone.
- **Context Retention:**
63.6% of users wanted the chatbot to remember previous emotional states.
- **Device Preference:**
72.7% preferred using a **mobile phone** to interact with chatbots.
- **Overall Satisfaction:**
71% of users rated their satisfaction with the chatbot at 4 or 5 out of 5.

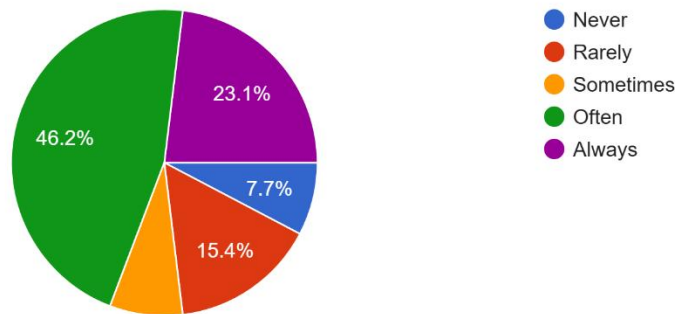
Feedback also indicated that users appreciated the emotional alignment of responses but suggested future improvements, such as adding deeper multi-turn conversation capabilities and more personalized follow-ups.

Age Group
13 responses



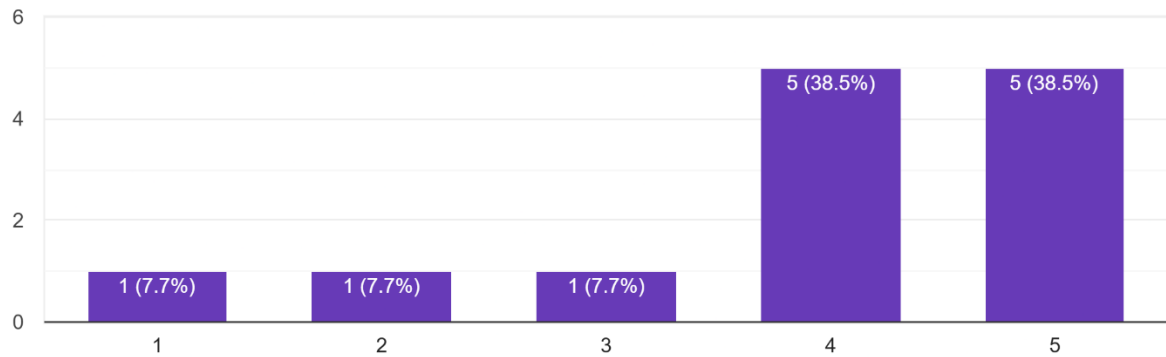
How often do you interact with chatbots?

13 responses



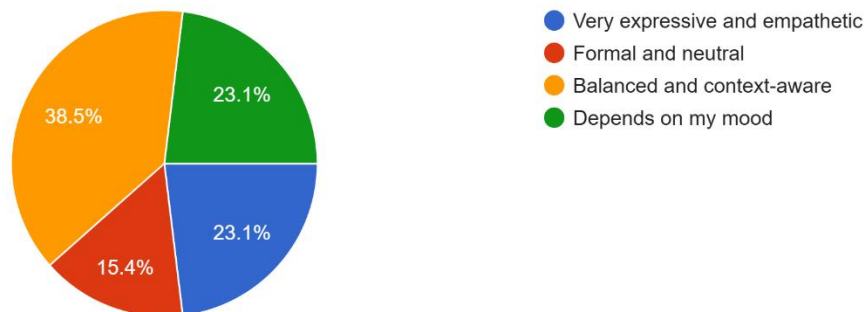
How important is it for a chatbot to recognize your emotions?

13 responses



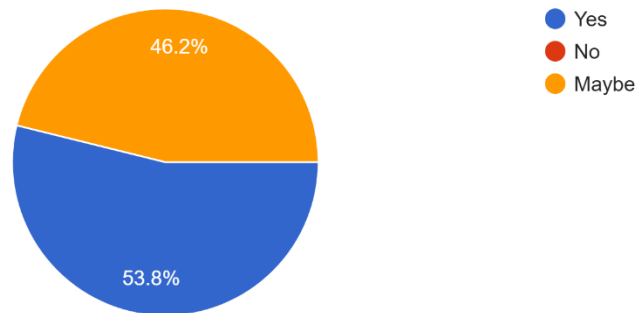
When chatting, what type of emotional tone do you prefer from a chatbot?

13 responses



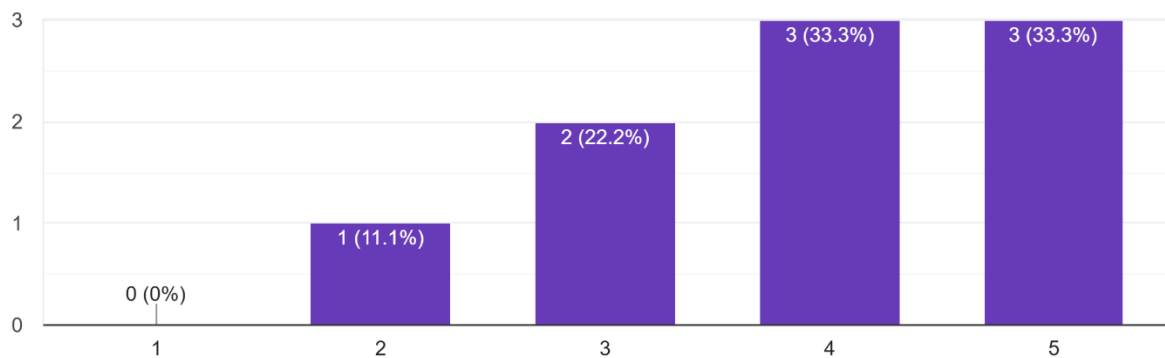
Would you like the chatbot to remember your previous emotions during a conversation (context retention)?

13 responses



How satisfied are you by using this ChatBot

9 responses



5.3 Discussion

The project achieved its goal of integrating **affective computing principles** into a chatbot capable of delivering emotionally aware responses.

Strengths:

- High classification accuracy and strong F1-scores across emotions.
- Positive user satisfaction with emotional detection and response appropriateness.
- Lightweight and accessible interaction through the command-line interface.

Limitations:

- Some emotional misclassifications between similar emotions (e.g., sadness and fear).
- Static, predefined responses instead of dynamically generated text.
- No persistent memory across multi-turn dialogues.

Future Work Recommendations:

- Integrate memory mechanisms to handle multi-turn conversations.
- Expand to more nuanced emotions (e.g., gratitude, shame) by retraining on larger datasets like GoEmotions.
- Upgrade from a command-line interface to a basic web or mobile-based conversational interface for better usability.

6. Conclusion and Future Work

6.1 Conclusion

This project successfully combined principles from **Human-Computer Interaction (HCI)** and **Natural Language Processing (NLP)** to build a **sentiment-aware chatbot** capable of detecting user emotions and providing contextually appropriate, empathetic responses.

The chatbot, developed using a fine-tuned **BERT-based model**, achieved high accuracy (88.32%) and strong macro F1 performance (88.25%) across key emotional categories (anger, fear, joy, sadness).

Usability testing through a **Google Forms survey** confirmed that the chatbot's emotional detection and responsiveness were positively received by users, with over **71% satisfaction rates**.

The use of a **simple command-line interface** allowed efficient interaction and evaluation without the overhead of complex GUIs. The project also integrated **error analysis** through confusion matrices, helping to identify emotional categories where improvements are needed.

Overall, the chatbot demonstrated robust emotional intelligence in its responses, validating the effectiveness of integrating affective computing techniques into real-world conversational agents.

6.2 Future Work

To further enhance the chatbot's capabilities and user experience, the following improvements are recommended:

- **Multi-Turn Context Management:**
Implement memory mechanisms to allow the chatbot to remember previous user inputs and emotions across multiple conversational turns, enabling more natural and coherent dialogues.
- **Expanded Emotional Range:**
Extend the model to recognize a broader set of emotions (e.g., gratitude, guilt, excitement) using larger datasets such as **GoEmotions** or **EmpatheticDialogues**.
- **Dynamic Response Generation:**
Move beyond static predefined responses by integrating a **response generation model** (e.g., fine-tuning GPT-2 or T5) to create more personalized and varied replies.
- **Interface Improvements:**
Develop a lightweight web-based or mobile app interface to improve accessibility and usability, especially since a majority of users preferred mobile-based interactions.
- **Advanced Affective Computing Techniques:**
Integrate continuous emotion tracking throughout sessions to adapt the chatbot's tone dynamically based on cumulative user emotions.
- **Personalization:**
Allow the chatbot to learn individual user preferences over time, adapting tone and response style according to prior interactions.

By pursuing these enhancements, the chatbot can evolve into a more emotionally intelligent and user-centric conversational agent, capable of supporting deeper and more meaningful interactions.

7. References

1. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Proceedings of NAACL-HLT. <https://arxiv.org/abs/1810.04805>
2. **International Survey on Emotion Antecedents and Reactions (ISEAR) Dataset.** Kaggle. <https://www.kaggle.com/datasets/faisalsanto007/isear-dataset>
3. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). *Transformers: State-of-the-Art Natural Language Processing*. Proceedings of the 2020 EMNLP: System Demonstrations. <https://arxiv.org/abs/1910.03771>

4. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011).**
Scikit-learn: Machine Learning in Python.
Journal of Machine Learning Research, 12, 2825–2830.
5. **Hunter, J. D. (2007).**
Matplotlib: A 2D Graphics Environment.
Computing in Science & Engineering, 9(3), 90–95.
6. **Waskom, M. L. (2021).**
Seaborn: Statistical Data Visualization.
Journal of Open Source Software, 6(60), 3021.
7. **Google Forms.**
<https://docs.google.com/forms/d/e/1FAIpQLSdcOLKUSwDreWpc3-2Q-9XuLWXbMo6YPy9Z1SnyBopOS-llew/viewform?usp=sharing>

8. Appendix

Appendix A: Tools and Technologies

Natural Language Processing (NLP)

- **Model:** BERT-base-uncased (Hugging Face Transformers)
- **Dataset:** ISEAR Dataset (Kaggle)
- **Libraries Used:**
 - Hugging Face Transformers
 - PyTorch
 - Scikit-learn
 - Matplotlib
 - Seaborn
- **Development Language:** Python 3.8+

Human-Computer Interaction (HCI)

- **Interface:** Command-line terminal (text-based interaction)
- **User Testing Platform:** Google Forms
- **Evaluation Method:**

- Likert scale ratings (1–5) for emotional appropriateness and chatbot usability
- Qualitative feedback collection via open-ended survey questions

Appendix B: Hardware and Software Environment

- **Operating System:** Windows 10 (or equivalent)
- **Processor:** Intel Core i5 or above
- **RAM:** Minimum 8 GB
- **Python Version:** 3.8+
- **Virtual Environment:** Recommended (venv or conda)

Appendix C: Key Evaluation Metrics

- Accuracy
- Precision, Recall, and F1-Score (Macro Average)
- Confusion Matrix
- User Satisfaction Survey Results
- Error Analysis through Seaborn Visualizations

Appendix D: Sample Chatbot Interactions (Terminal)

Examples of chatbot conversation include:

- User input expressing sadness → Chatbot providing empathetic supportive responses.
- User expressing joy → Chatbot responding with congratulatory excitement.
- User describing fear → Chatbot offering reassurance.

```
> My best friend moved to another country, and I miss her so much. I feel really lonely.
Chatbot: Sending virtual hugs. 🤗 (Detected emotion: sadness)
> I found out I got a promotion at work today! I'm absolutely ecstatic and so proud of myself.
Chatbot: Yay! I'm happy to hear that! 🎉 (Detected emotion: joy)
> Someone stole my wallet on the train. I'm furious and feel so violated.
Chatbot: I'm sorry you're feeling angry. 😡 (Detected emotion: anger)
> Hearing that strange noise downstairs in the middle of the night filled me with intense fear.
Chatbot: Stay strong! You got this. 🦊 (Detected emotion: fear)
> My neighbor helped me fix my car when it broke down. I'm so grateful for their kindness.
Chatbot: Yay! I'm happy to hear that! 🤗 (Detected emotion: joy)
> After all the hard work, seeing the final results made me feel a sense of deep satisfaction. Someone stole my wallet on the train. I'm furious and feel so violated.
Chatbot: I'm sorry you're feeling angry. 😡 (Detected emotion: anger)
> Hearing that strange noise downstairs in the middle of the night filled me with intense fear.
Chatbot: Stay strong! You got this. 🦊 (Detected emotion: fear)
> 
```