

# TheraBot: An Emotionally-Aware AI Mental Health Companion Using Joint Emotion-Sarcasm Detection and Generative Response Modeling

Sri Sai Charan Yarlagadda

Northeastern University

Email: yarlagadda.sr@northeastern.edu

Shivram Nekkanti

Northeastern University

Email: nekkanti.sh@northeastern.edu

**Abstract**—Artificial intelligence holds great promise in scaling mental health support through conversational agents. However, most existing systems struggle with understanding the emotional nuance of user inputs, especially when sarcasm is present. This often results in emotionally misaligned or ineffective responses, which can be detrimental in sensitive contexts like mental health.

We introduce TheraBot, an emotionally-aware AI companion that incorporates joint emotion and sarcasm detection to generate empathetic responses. Our system employs a BERTweet-based emotion classifier fine-tuned on a clustered GoEmotions dataset and a RoBERTa-based sarcasm detector trained on the SARC dataset. The outputs are processed through an emotion adjustment logic, which refines the emotion label by incorporating sarcasm signals. This refined emotion context, along with chat history, is used to condition a distilGPT2-based response generator.

TheraBot is deployed through a Streamlit interface for real-time interaction. Our evaluation demonstrates improved emotional alignment, especially in sarcastic scenarios, showcasing the potential of TheraBot in delivering emotionally intelligent responses. This work contributes to the development of AI companions better suited for emotionally nuanced human communication.

## I. INTRODUCTION

The growing global mental health crisis has increased interest in AI-powered tools that can provide scalable and accessible support. Conversational agents, including chatbots and virtual companions, are increasingly adopted to address the need for low-cost mental health interventions. However, most of these systems lack a critical component—emotional intelligence. They often respond in generic or misaligned ways, failing to understand the emotional tone or intent of user inputs, particularly when sarcasm is involved.

Sarcasm, by its nature, can invert the intended meaning of a message and cause sentiment misclassification in emotion detection models. This misalignment can be especially harmful in mental health settings, where empathetic and accurate interpretation of emotional cues is crucial. Addressing this, we propose **TheraBot**, an emotionally-aware mental health companion that combines both emotion and sarcasm detection to improve contextual understanding and generate more appropriate responses.

Our contributions are threefold. First, we fine-tune a BERTweet model for emotion classification using a curated 10-class version of the GoEmotions dataset. Second, we train

a RoBERTa-based model on the SARC dataset for sarcasm detection. We introduce a logic-based module that adjusts the detected emotion when sarcasm is present. Finally, we integrate a distilGPT2-based generative model that produces context-sensitive replies conditioned on the adjusted emotion and prior conversation history.

TheraBot is deployed as an interactive chatbot through Streamlit, enabling real-time communication. Our results indicate that the system provides more emotionally aligned responses compared to baseline models, especially in emotionally ambiguous or sarcastic situations. This paper presents a step toward building emotionally intelligent AI companions for mental health and related applications.

## II. LITERATURE SURVEY

Understanding the landscape of research in emotion detection, sarcasm identification, and generative modeling is critical to contextualize our contributions. This literature survey highlights key works that have inspired the design of TheraBot, including advances in transformer models, emotion-aware datasets, and applications in conversational AI. Below are major components that form the backbone of our system:

- **Emotion Detection in NLP:** The GoEmotions dataset [1] offers a rich taxonomy of 58 emotion categories annotated on Reddit comments. BERT-based models, particularly BERTweet [2], pretrained on tweets, have shown strong performance on emotion classification tasks in conversational contexts.
- **Sarcasm Detection:** Sarcasm detection is essential for interpreting the true emotional tone of user inputs. The SARC dataset [3] provides a large corpus of sarcastic and non-sarcastic Reddit comments. RoBERTa [4], when fine-tuned on this dataset, is effective in identifying sarcasm by leveraging contextual embeddings.
- **Generative Conversational Models:** Generative models like GPT-2 [5] and its lighter variant distilGPT2 [6] have revolutionized text generation by producing coherent and human-like responses. BlenderBot [7] further pushes conversational capabilities through fine-tuning on dialogue datasets.
- **Mental Health Chatbots:** Early systems like ELIZA [8] demonstrated basic pattern-matching responses. Modern

bots such as Woebot [9] and Replika employ rule-based and neural approaches, respectively, to provide supportive interactions. However, they generally lack explicit emotion-sarcasm understanding, which limits emotional alignment.

### III. OBJECTIVES

TheraBot is designed to bridge the gap between emotionally aware AI and practical mental health conversational agents. The following objectives outline the key pillars that structure the development and deployment of the system:

#### A. Promote Emotionally Aware Interactions

The primary objective of TheraBot is to recognize and respond to the emotional state of users in a meaningful way. This involves detecting nuanced emotional cues from text, including informal and social media language, and tailoring responses that align with the user's feelings. By doing so, the chatbot can create more empathetic and supportive interactions, especially important in sensitive conversations related to mental health.

#### B. Incorporate Sarcasm Awareness

Sarcasm is a linguistic challenge in NLP that can significantly alter the sentiment conveyed by a message. TheraBot includes a dedicated sarcasm detection module that ensures sarcastic expressions do not mislead the emotion classifier. Recognizing sarcasm helps reinterpret user intentions and provides a more accurate emotional understanding, which is crucial in mental health applications where misinterpretation can be detrimental.

#### C. Enable Nuanced Response Generation

A key function of TheraBot is to generate responses that are emotionally aligned and contextually appropriate. This is achieved by conditioning a generative language model on detected emotions, sarcasm flags, and recent conversation history. By doing this, TheraBot avoids generic replies and instead delivers responses that are sensitive to the user's emotional needs.

#### D. Facilitate Real-Time Usability

The system is deployed through a web-based chatbot interface built with Streamlit, allowing users to interact in real-time. This objective ensures the system is accessible and responsive, simulating the immediacy of a live conversation. Real-time usability is critical to fostering trust and maintaining user engagement during emotional disclosures.

#### E. Ensure Ethical Design Principles

Ethics play a foundational role in TheraBot's development. This objective includes ensuring user privacy, avoiding bias in training data, and maintaining transparency in how the model interprets and responds to inputs. TheraBot is not intended to replace professional therapy but to offer supportive engagement, and it is designed with this boundary clearly in mind.

### F. Support Extensibility

THERABOT is built with a modular architecture, allowing easy integration of additional features in the future. These may include new emotion categories, multimodal inputs like voice or facial recognition, and support for multilingual interactions. This flexibility ensures that TheraBot can evolve to meet broader user needs and adapt to new research in AI and mental health.

## IV. DATASET AND PREPROCESSING

To develop an emotionally intelligent chatbot like TheraBot, we utilize a combination of existing public datasets and custom-annotated data. This section presents the data sources employed and the preprocessing steps taken to ensure effective training and inference.

#### A. GoEmotions Dataset (Clustered)

The GoEmotions dataset contains over 58,000 Reddit comments annotated with 28 emotion labels. For TheraBot, we reorganized these into 10 broader emotional categories to strike a balance between granularity and model performance. The clustering helped mitigate class imbalance and improved the generalizability of the emotion classifier. The dataset is particularly well-suited for capturing emotion in online discourse due to its origin in Reddit comments. We applied light cleaning, punctuation normalization, and processed the data using BERTweet's tokenizer to preserve social media-specific language.

```
[
  {
    "text": "I can't believe how amazing today was! 🥳🥳",
    "emotion": "joy",
    "emotion_id": 5
  },
  {
    "text": "Why does this always happen to me? 😞😞",
    "emotion": "confusion",
    "emotion_id": 1
  }
]
```

Fig. 1: A sample data point of the GoEmotions dataset

#### B. SARC Dataset

The Self-Annotated Reddit Corpus (SARC) comprises over 200k+ comments labeled for sarcasm. We selected a balanced subset of sarcastic and non-sarcastic samples to avoid class skew. The dataset spans a wide range of topics and linguistic expressions, making it ideal for training a robust sarcasm detection model. Preprocessing included removal of links, special tokens, and tokenization using the RoBERTa tokenizer. Sarcasm detection benefits from such a large and diverse dataset by learning subtle patterns of ironic expression, particularly relevant in emotionally complex conversations.

```

16 [
17   {
18     "text": "Oh great, another Monday. Just what I needed.",
19     "label": "sarcastic",
20     "label_id": 1
21   },
22   {
23     "text": "Sure, because staying up all night is totally healthy.",
24     "label": "sarcastic",
25     "label_id": 1
26   }
27 ]

```

Fig. 2: A sample data point of the SARC dataset

### C. Psychology\_10k (Custom Dataset)

The Psychology\_10k dataset was manually curated from public mental health discussions and annotated with both emotion and sarcasm labels. With around 10,000 entries, it provides crucial real-world conversational data that reflects the subtleties found in mental health communication. It enabled joint modeling of emotional tone and sarcasm in the same utterance. This dataset strengthens the model’s ability to reason about emotion in psychologically sensitive and context-rich dialogue. In order to make the data compatible with our model input, we used our fine-tuned models to generate the emotion and sarcasm labels for each user input and then added them to the dataset.

### D. Preprocessing

All datasets underwent uniform preprocessing steps to standardize input format and facilitate training. These steps include:

- **Text Cleaning:** This involved removing hyperlinks, user mentions, HTML entities, and repeated punctuation. These artifacts introduce noise and may distract the models from capturing genuine emotional or sarcastic signals.
- **Emoji Normalization:** The deemojize function from the emoji library in python was used to convert the emojis into text.
- **Tokenization:** Inputs were tokenized using model-specific tokenizers — BERTweet for emotion and RoBERTa for sarcasm. These tokenizers retain slang, emojis, and informal syntax typical in social dialogue.
- **Label Merging:** When training the joint classifier, both emotion and sarcasm labels were unified in a single data structure. This allowed the model to simultaneously learn from dual aspects of the emotional context.
- **Prompt Engineering:** For the response generation model, we constructed inputs that included the inferred emotion, sarcasm flag, and prior conversation history. This structured prompt helped distilGPT2 produce context-sensitive and emotionally aligned replies.

## V. SYSTEM ARCHITECTURE

The TheraBot system is structured as a modular pipeline that takes in user input, interprets emotional tone and sarcasm, adjusts for emotional nuance, and generates a human-like, empathetic response. The architecture is designed for flexibility, extensibility, and real-time interaction through a web

interface. Below is an illustration and detailed explanation of each component involved in the end-to-end workflow.

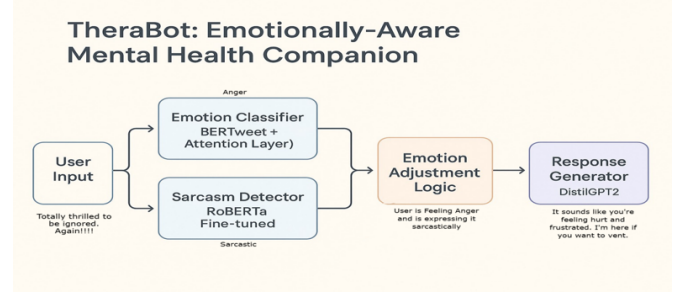


Fig. 3: TheraBot System Architecture: Emotion and Sarcasm Driven Response Pipeline

### A. User Input

Users interact with TheraBot by entering a message or statement into the chat interface. These messages often contain emotional cues, colloquialisms, or sarcastic undertones that need careful interpretation. User input serves as the entry point for the inference pipeline, triggering emotion and sarcasm analysis in parallel.

### B. Emotion Classifier (BERTweet + Attention Layer)

The emotion classifier is a fine-tuned BERTweet model enhanced with an attention pooling layer. It classifies user input into one of ten predefined emotional categories (e.g., anger, joy, sadness). Attention pooling allows the model to prioritize emotionally charged tokens, improving classification accuracy. This layer ensures that fine-grained emotional content is captured even in brief, informal user messages.

### C. Sarcasm Detector (RoBERTa Fine-Tuned)

The input is simultaneously processed through a RoBERTa-based model fine-tuned on the SARC dataset and Twitter Sarcasm Corpus. It predicts whether the input message contains sarcasm, helping disambiguate literal sentiment from intended meaning. This parallel detection pipeline ensures that sarcasm detection is decoupled from emotion classification, increasing modularity and flexibility.

### D. Emotion Adjustment Logic

The emotion adjustment module takes both emotion and sarcasm predictions into account. If sarcasm is detected, the system applies rule-based heuristics to adjust the emotion label accordingly. This step is crucial to avoid mismatches between user intent and system-generated responses, improving the emotional sensitivity of the interaction.

### E. Response Generator (distilGPT2 with Conversation Context)

The adjusted emotion label and the recent conversation history are then combined into a structured prompt. This prompt is passed to a distilGPT2 model, fine-tuned for emotionally aware conversation generation. The model generates responses

that are empathetic, contextually relevant, and emotionally aligned, closing the inference loop back to the user in a natural dialogue flow.

#### F. UI Output (Flask Chatbot Interface)

The final response is rendered on the Flask-based user interface. The UI handles multi-turn conversation memory and ensures that both user messages and TheraBot responses are displayed clearly. By providing an accessible and visually appealing frontend, the application enhances overall user engagement and trust.

### VI. MODEL TRAINING

The training of TheraBot involves a multi-stage process designed to optimize three primary components: the emotion classifier, the sarcasm detector, and the generative response model. Each model was trained independently using domain-specific datasets and fine-tuned hyperparameters to improve their respective performance.

#### Emotion Classifier (BERTweet + Attention)

The emotion classifier is based on BERTweet, a transformer model pre-trained on 850M English tweets. This makes it well-suited for informal, emotionally expressive language common in user interactions. We fine-tuned the model on a clustered 10-class version of the GoEmotions dataset.

To enhance its capability to focus on emotionally relevant parts of a sentence, we added an attention pooling layer after the final hidden state. This layer computes a weighted sum of token embeddings based on learned attention weights, allowing the model to prioritize emotionally charged words.

Training was conducted for 4 epochs using the AdamW optimizer with a learning rate of  $2e-5$  and batch size of 16. Cross-entropy loss was used due to the multi-class classification nature of the task. We used early stopping based on validation F1-score to prevent overfitting.

#### Sarcasm Detector (RoBERTa)

The sarcasm detector was built using RoBERTa-base, a robust variant of BERT optimized for pretraining. We fine-tuned the model on a balanced subset of the SARC dataset, ensuring equal representation of sarcastic and non-sarcastic samples.

Sarcasm is inherently difficult to detect due to its dependency on subtle contextual cues, tone, and irony. To address this, we incorporated context-aware fine-tuning where conversational threads were preserved to provide background for each comment. Training spanned 3 epochs with a learning rate of  $3e-5$  and batch size of 32. Binary cross-entropy loss was used as sarcasm detection is a binary classification task. Accuracy and F1-score were used to evaluate performance, and the model achieved high precision in detecting nuanced sarcastic intent, especially in short, sentimentally ambiguous messages.

#### Joint Inference Module

This component acts as a semantic bridge between the emotion and sarcasm detectors. Implemented as a rule-based logic module, it evaluates the detected emotion in light of the sarcasm prediction. If sarcasm is detected, the original emotion label is adjusted based on heuristics derived from empirical analysis.

For example, user input originally classified as "joy" may be flipped to "frustration" or "confusion" if the sarcasm detector confidently flags it as sarcastic. These rules were constructed through observation of overlapping emotional patterns in sarcastic expressions from the Psychology\_10k dataset. The adjustment logic not only improves emotional alignment but also prevents misleading response generation by accurately representing the user's true emotional tone. While rule-based in nature, the logic was built with the flexibility to support future enhancements via learned heuristics or reinforcement feedback.

#### Response Generator (distilGPT2)

The response generator is a fine-tuned distilGPT2 model trained on prompt-response pairs constructed from Psychology\_10k and synthetic conversation data. Prompts included user message, detected emotion, sarcasm label, and recent chat history.

We used a learning rate of  $5e-5$  with a batch size of 4 and trained for 5 epochs. The model was optimized using the Adam optimizer and trained with a maximum sequence length of 256 tokens. We applied top-k sampling and temperature scaling ( $T=0.7$ ) during inference to maintain diversity and control repetition.

By conditioning on both emotion and sarcasm, the model learns to generate contextually relevant and empathetic responses. Performance was qualitatively evaluated through sample outputs and human feedback, confirming the fluency and emotional alignment of generated replies.

### VII. EVALUATION AND RESULTS

To assess TheraBot's performance, we evaluated both the emotion classification model and the sarcasm detection model using standard classification metrics such as accuracy, precision, recall, and F1-score on the validation sets. Below we present the evaluation results in tabular form.

## Emotion Classification Evaluation

TABLE I: Classification Report: Emotion Classifier (Test Set)

Class ID	Precision	Recall	F1-Score	Support
0	0.66	0.56	0.61	150
1	0.68	0.79	0.73	150
2	0.80	0.77	0.78	150
3	0.60	0.60	0.60	150
4	0.91	0.97	0.94	150
5	0.84	0.91	0.87	150
6	0.70	0.65	0.67	150
7	0.81	0.78	0.79	150
8	0.52	0.45	0.48	150
9	0.71	0.80	0.75	150
Accuracy			0.73	1500
Macro Avg	0.72	0.73	0.72	1500
Weighted Avg	0.72	0.73	0.72	1500

## Sarcasm Detection Evaluation

TABLE II: Classification Report: Sarcasm Detector (Test Set)

Label	Precision	Recall	F1-Score	Support
Not Sarcasm	0.76	0.70	0.73	1000
Sarcasm	0.72	0.78	0.75	1000
Accuracy			0.74	2000
Macro Avg	0.74	0.74	0.74	2000
Weighted Avg	0.74	0.74	0.74	2000

## Emotion Flip via Sarcasm Detection

TheraBot’s Emotion Adjustment Logic plays a vital role in reinterpreting emotional context. For instance, if a user expresses ”Totally thrilled to be ignored. Again!” the emotion classifier might detect ”joy.” However, once sarcasm is detected, this emotion is adjusted to ”frustration” or ”confusion” based on predefined rules. This flip ensures the generated response remains contextually and emotionally aligned.

In another example, consider the input: ”Oh sure, let’s just cancel everything last minute again!”. The emotion classifier initially identifies the emotion as **anger** with a confidence score of 0.442. Simultaneously, the sarcasm detector confidently flags the input as **sarcastic** with a confidence score of 0.993. As a result, the joint inference system activates the Emotion Adjustment Logic, overriding the default emotional tag based on sarcasm presence. The model reinterprets the sentiment as potentially expressing **disappointment**, **resentment**, or **passive-aggressive frustration**. This realignment not only prevents emotionally tone-deaf responses but also ensures that the generative model’s output is more supportive and empathetically grounded.

## Sample Interaction (Emotion-Aware Response)

**User Input:** I’m really scared about my job interview tomorrow!

**Generated Response:**

purple It’s completely normal to feel nervous before a big interview, but it’s important to remember that it’s a normal part of life. Let’s work together to develop coping strategies

and build a positive relationship with your job. We can also explore ways to prepare for the interview by practicing relaxation techniques and practicing positive self-talk. Remember, it’s okay to take risks when you feel nervous and anxious. Have you considered talking to your supervisor about your concerns? It may also be helpful to practice self-

## VIII. THERABOT APPLICATION

TheraBot is deployed as a web-based application, featuring a Flask-powered backend and a custom HTML/CSS frontend. The system integrates backend inference capabilities with a lightweight, intuitive user interface to enable seamless interaction.

## Interface Overview

The frontend is developed using standard HTML, CSS, and JavaScript technologies, presenting a clean and responsive chat layout (Figure 1). User inputs are collected through a styled text field, and responses are dynamically displayed on the same page without requiring reloads. Emotionally aware content is highlighted using contextual emoji markers such as purple. The interface includes two primary buttons—`Send` for message submission and `Reset` to clear the conversation history.

## Features

- Emotion-aware and sarcasm-aware conversation flow
- Maintains recent chat history to provide contextually consistent responses
- Dynamic web interface with fast backend communication via Flask routes
- Designed to be modular and extensible for future UI enhancements

## Backend Architecture

The backend server is implemented using Flask and integrates:

- Fine-tuned BERTweet model for emotion classification
- Fine-tuned RoBERTa model for sarcasm detection
- Emotion adjustment module for sarcasm-aware emotion refinement
- Fine-tuned distilGPT2 model for emotionally grounded response generation

The backend handles incoming user messages through HTTP POST requests, processes joint emotion and sarcasm inference, adjusts the detected emotions if necessary, and generates an appropriate reply. The response is sent back to the frontend in real-time for display.



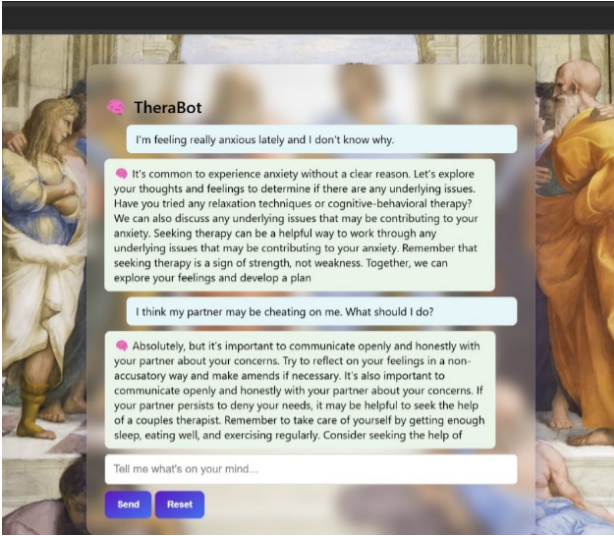


Fig. 4: TheraBot UI – Multi-turn, Emotion-Aware Interaction

## IX. LIMITATIONS AND ETHICAL CONSIDERATIONS

While TheraBot demonstrates significant progress toward empathetic AI, it also comes with limitations and ethical concerns:

- **Bias in Training Data:** The underlying models may inherit biases from Reddit and mental health forums, affecting fairness.
- **Positive Prompt Misalignment:** Due to the response generator being fine-tuned predominantly on a psychology-oriented dataset, TheraBot tends to produce therapeutic or overly serious responses even for light-hearted or positive prompts. Example: When given the user input *"I'm excited about my vacation next week!"*, instead of generating a simple celebratory response, TheraBot might respond with something like *"I am so sorry that you are on a vacation"*. This shows that the model occasionally frames positive emotions in a therapeutic lens rather than matching the natural tone.
- **Not a Replacement for Therapy:** TheraBot is intended for supportive conversations and not for diagnosing or treating mental health conditions.
- **Data Privacy:** All inputs are processed locally or through server memory without user authentication, and no messages are stored. However, production deployment would require robust privacy safeguards.
- **Overdependence Risk:** Users may become emotionally dependent on the chatbot, mistaking it for a human or therapeutic substitute.

## X. INDIVIDUAL CONTRIBUTIONS

### Sri Sai Charan Yarlagadda:

- Led the preprocessing of the SARC dataset and the Twitter Sarcasm Corpus, ensuring data cleanliness, balance, and proper tokenization for sarcasm detection.
- Developed and fine-tuned the Sarcasm Detection Model using cardiffnlp/twitter-roberta-base, optimizing it for binary classification of sarcastic and non-sarcastic inputs.
- Designed and implemented the web app for TheraBot using Flask, HTML/CSS, creating a clean and user-friendly real-time chat application experience.

### Shivram Nekkanti:

- Handled the preprocessing of the GoEmotions dataset, including emotion label clustering, multi-label filtering, and tokenization for single-label classification tasks.
- Created and fine-tuned the Emotion Classifier Model using vinai/bertweet-base with an added attention pooling layer for improved emotional sensitivity.
- Fine-tuned the DistilGPT2 Generative Model on the custom Psychology\_10k dataset with emotion and sarcasm conditioning to enable emotionally aligned and context-aware response generation in TheraBot.

## XI. CONCLUSION AND FUTURE WORK

In this work, we present **TheraBot**, a mental health companion chatbot that integrates state-of-the-art emotion recognition, sarcasm detection, and response generation capabilities. The system aims to bridge the gap between emotional nuance and machine understanding by using a multi-model pipeline built on top of BERTweet, RoBERTa, and distilGPT2. By jointly analyzing emotional tone and sarcastic cues in user inputs, TheraBot generates responses that are emotionally aligned, supportive, and tailored to each individual interaction.

Our evaluations, including validation accuracy and classification reports, demonstrate that the emotion and sarcasm classifiers perform well across diverse user expressions. The emotion adjustment logic layer further enhances this alignment by flipping misinterpreted emotional tones when sarcasm is detected, resulting in more accurate and empathetic responses. Example conversations in the application interface highlight the potential of TheraBot to offer meaningful and comforting feedback in real-time. Additionally, our modular architecture and intuitive Streamlit interface provide a strong foundation for real-world deployment and future expansion.

Beyond technical performance, TheraBot encourages exploration into emotionally intelligent systems that prioritize user experience, particularly for mental health support. While it does not replace licensed therapy, it offers a conversational safety net for individuals seeking low-barrier emotional engagement.

Looking forward, several areas of enhancement can make TheraBot more impactful:

- **Voice Input Integration:** To improve accessibility and user engagement, we plan to introduce voice-to-text capabilities that allow users to speak their thoughts directly to the system.

- **Broader Emotion Taxonomies:** We aim to expand the emotional label space to include compound and nuanced emotions, such as guilt, regret, or awe, that are currently underrepresented in our dataset.
- **Clinical Trials and Real-World Validation:** Our ultimate goal is to evaluate TheraBot in collaboration with licensed mental health professionals through controlled studies and user surveys, validating its therapeutic value and identifying areas for safety and improvement.
- **User Feedback System:** A thumbs-up/thumbs-down feedback feature will be added to allow users to rate TheraBot’s responses in real time, facilitating continuous model refinement and alignment with user expectations.
- **Use of Larger Language Models:** We plan to experiment with more powerful open-weight LLMs such as Mistral-7B, Mixtral, and TinyLlama for both classification and generation tasks. These models offer improved fluency, deeper contextual understanding, and greater scalability.

In summary, TheraBot sets the stage for emotionally intelligent AI systems that go beyond intent classification by embracing the complexity of human communication. Its modular design, empirical results, and emotional grounding make it a promising framework for future AI companions in mental wellness and beyond.

## REFERENCES

- [1] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, “Goemotions: A dataset of fine-grained emotions,” *arXiv preprint arXiv:2005.00547*, 2020.
- [2] D. Q. Nguyen, T. Vu, and A. T. Nguyen, “Bertweet: A pre-trained language model for english tweets,” *arXiv preprint arXiv:2005.10200*, 2020.
- [3] M. Khodak, N. Saunshi, and K. Vodrahalli, “A large self-annotated corpus for sarcasm,” in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.
- [4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, 2019.
- [6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.
- [7] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y.-L. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith *et al.*, “Recipes for building an open-domain chatbot,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021, pp. 300–325.
- [8] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [9] K. K. Fitzpatrick, A. Darcy, and M. Vierhile, “Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial,” *JMIR Mental Health*, vol. 4, no. 2, p. e19, 2017.

## XII. APPENDIX: HOW TO RUN THE PROJECT

### Prerequisites

- Python 3.7+
- CUDA-compatible GPU (recommended) or CPU

### Setup Instructions

- 1) Create and activate a virtual environment (recommended):
  - `python -m venv venv`
  - On Windows: `venv\Scripts\activate`
  - On Unix or MacOS: `source venv/bin/activate`
- 2) Install the required Python packages:
  - `pip install -r requirements.txt`

### Required Models

The application requires pre-trained models to be available locally:

Emotion detection model (located in `models/Emotion_model/`)  
 Sarcasm detection model (located in `models/Sarcasm_detection_model/`)  
 Ensure all model files are present in their respective directories before starting the application.

### Running the Application

- 1) Navigate to the application directory:
  - `cd therabot_app`
- 2) Start the Flask server:
  - `python app.py`
- 3) Open your web browser and visit:
  - `http://localhost:5000`

### Usage Instructions

- Open the web interface in your browser.
- Type your message in the chat input.
- The bot will analyze your message for emotions and sarcasm, generate an appropriate response, and display it.
- Use the reset button to clear the chat history.

### Technical Details

The application stack includes:

- Flask for the web server
- PyTorch for deep learning models
- HuggingFace Transformers library for NLP tasks
- Custom attention-based emotion classification
- Fine-tuned DistilGPT2 for response generation

### Note

The model paths in `app.py` are currently set to absolute paths. Modify the following variables to match your system’s directory structure if needed.