Empathic Companion: An AI-Powered Chatbot for Emotional Support

Group Name: Group A
Group Members:

- Chadrick Clarke (c0929059)
- Sri Padma Paanidhar (c0924368)
- Sourab Botu (c0928050)
- Sarala Sunkara (c0924370)
- Olumide Emmanuel Akinyemi (c0924273)

Introduction to Empathic Companion

Technological Growth and societal impact:

- The technological advancements has progressed from internet and created a new from of scientific field called Artificial intelligence

- Many applications of AI like GPT, chatbots, smart home devices like Alexa, Google, Siri are being adapted by everyone very easily.(Lakhlani 2024)

- The internet had also enabled a fast paced and hectic lifestyle. Which has been seeping into work-life balance of many humans

- This new changes has affected the mental health. Forbes stated "*Young adults ages 18 the U.S. experience the highest rates of mental illness (36.2%), followed by those ages 26 to (29.4%) and adults ages 50 and over (13.9%).*"

Mental Health Challenges and AI Solutions

- Even with the high prevalence of mental health issues. 57% of the affected do not receive any counselling (Stephenson, 2023)

- The traditional support is lacking to provide the required help the individual needs.

- AI can be utilized to fill this gap by effectively communicating to user and give a non judgemental emotional support in the form of a Chatbot.

- This AI based chatbot will act as preliminary form of support and will offers imm assistance whenever needed.

Problem Statement

Meeting Unmet Needs: "Empathic Companion"

Nearly 600,000 Canadians (14%) report fair or poor mental well-being (Statista, 2024), often lacking access to counseling and support. Traditional therapy faces barriers like cost, scheduling, and stigma. There's a need for a low-barrier, accessible solution providing safe, nonjudgmental, and immediate preliminary help.

The "Empathic Companion: An Interactive Chatbot for Emotional Support" aims to address these issues. Using AI and NLP, the chatbot offers compassionate conversations and empathetic responses, providing immediate, nonjudgmental support and enhancing users' overall mental well-being.

Methodology

Approach to built a chatbot:

- Data Collection
- Data Preprocessing
- Sentiment and Emotion Detection
- Chatbot Development
- Testing and Evaluation

Technologies Used: BERT, GPT-2, TensorFlow, Keras, NLTK, Torch

Data Collection

Data Sources and Project Design

- **Initial Data Source**: Raw Tweets
    - **Size**: 40,000 records
    - **Features**: tweet_id, sentiment, content
    - **Reason**: Variability and complexity of emotions
- **Challenges**:
    - Difficulties in real-time data collection and processing due to **complexity and non-flexibility data**
    - Limited project timeframe of 12 weeks

- **Resolution**:
    - **Dataset Selection**: Used a pre-existing dataset of tweets
    - **Additional Dataset**: Added Cornell Movie-Dialogue Corpus
        - **Size**: 304,713 records
        - **Features**: conversation_id, text, speaker, reply-to
        - **Purpose**: Improved chatbot response quality
- **Outcome**: Balanced data sources to address complexity and enhance model perfo

Data Preprocessing

Wrangling

- **Methods**:
  - Removing Digits
  - Removing punctuations
  - Cleaning text
  - Word lengthening
  - Tokenize to remove stopwords

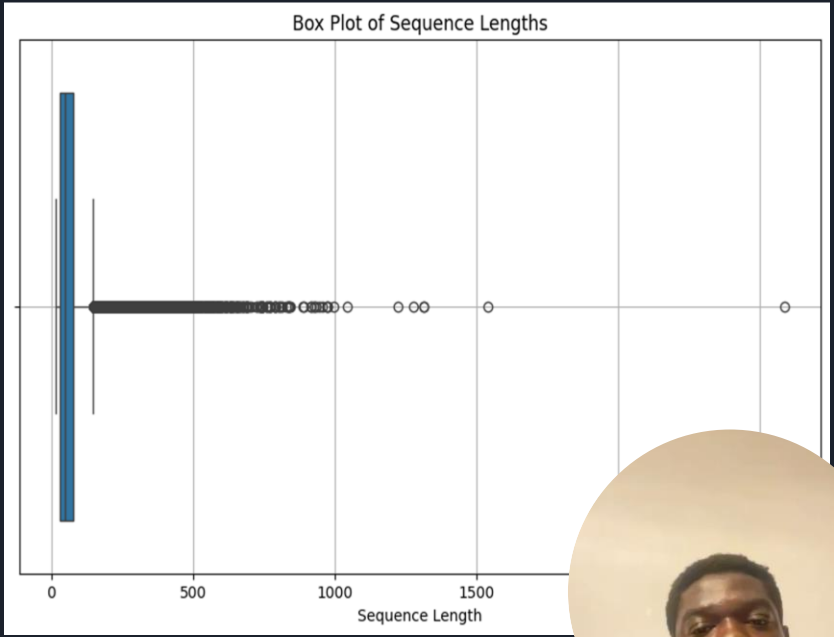
Processing Steps:

- **Missing Values**: No missing values in the dataset
- **Text Cleaning**: Removed HTML tags, special characters, digits, and stopwords
- **Emojis**: Retained for emotional context
- **Emotion Reclassification**: Used Hugging Face transformer library for accurate emotion labels
- **User-Defined Functions**: replace_contractions, preprocessing
- **Text Cleaning**: Removed punctuations, digits, emojis, special characters
- **Contraction Replacement**: e.g., "who’ve" → "who have"
- **Tokenization and Lemmatization**: Applied using spaCy (nlp(), lemma_, is_alpha)

Outlier Detection

Findings:

- Significant word sequence lengths varied from ~200 to ~2700.
- Outliers were detected beyond a word sequence length of 147.5.
- Middle 50% of word sequences had a length within an IQR of 46.



Sentiment and Emotion Detection Model Development

Sentiment Analysis with Vader: Utilized Vader (Valence Aware Dictionary and sEntiment Reasoner) from the NLTK library to extract sentiment scores.

The design concept for this component is based on principles of extracting the sentiment from a text and correctly classifying the emotion of the text.

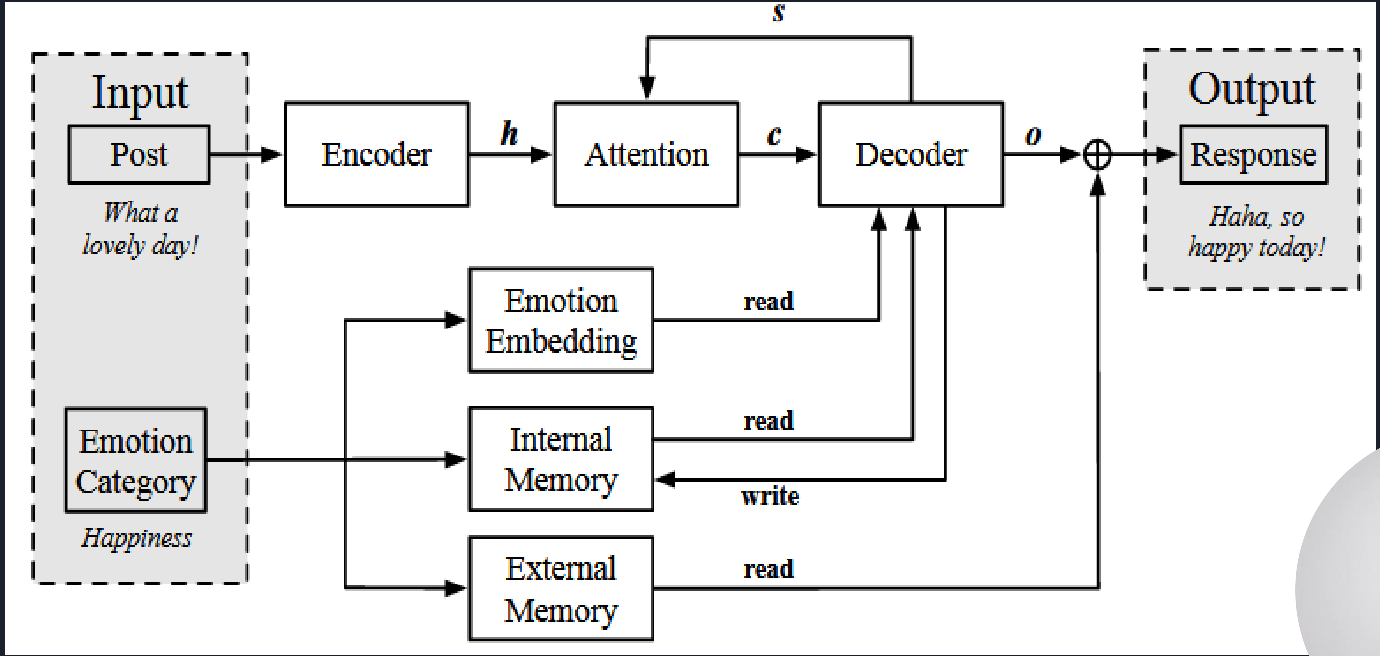
Emotion Classification:

- Algorithms Evaluated:

Models	Accuracy
Naive-Bayes	60%
Random Forest	64%
Support Vector Machine	67%

- Data Transformation: Applied TFIDFVectorizer to convert text to numerical format

Chatbot Architecture (transformer-based)



Chatbot Architecture (non-transformer based)

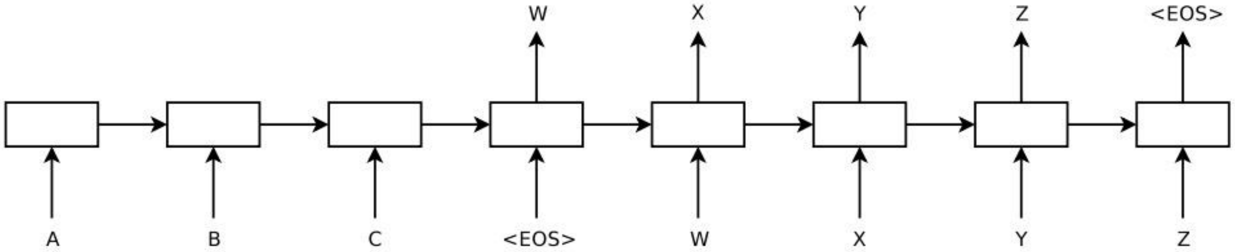
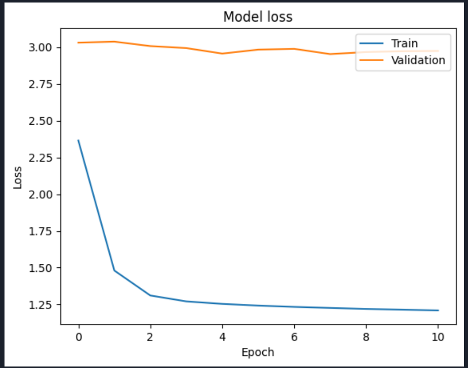


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads th input sentence in reverse, because doing so introduces many short term dependencies in the data that mak optimization problem much easier.

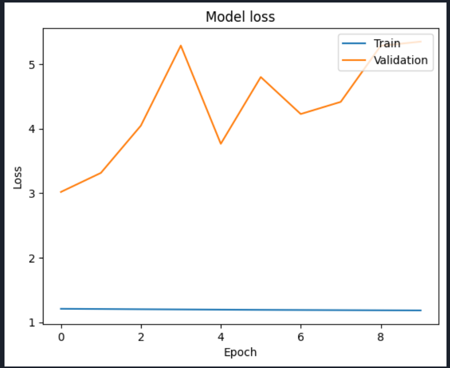
LSTM-Chatbot Architecture (non-transformer based)

1. **Model Overview:**
   - **LSTM (Long Short-Term Memory):** Used for sequence-to-sequence tasks like chatbots and language translation, accessible via TensorFlow and Keras.
2. **Data Preparation:**
   - **Tokenization:** Utilized Tokenizer from Keras to convert text into numerical tokens,
   - **Padding:** Applied `pad_sequences` to ensure uniform sequence length, adjusting `maxlen` to balance dimensionality and memory usage.
3. **Encoder and Decoder Setup:**
   - **Embedding Layer**
   - **LSTM Configuration**
4. **Attention Layer Integration:**
   - **Purpose:** Addressed fixed-length vector limitations of the encoder-decoder architecture to improve model coherence and response generation.
5. **Model Compilation and Training:**
   - **Compilation:** Optimizer (RMSprop), loss function (SparseCategoricalCrossentropy), and metrics (accuracy) defined.
   - **Training:** Model fitted with inputs, validation, and callback parameters for performance evaluation.

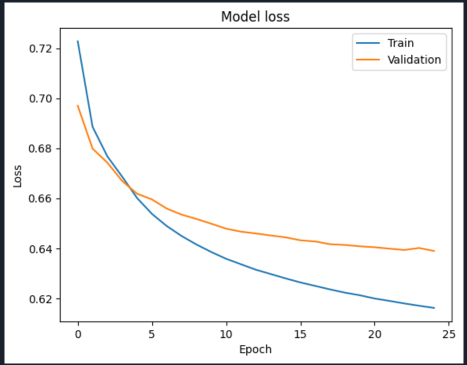
Model Training



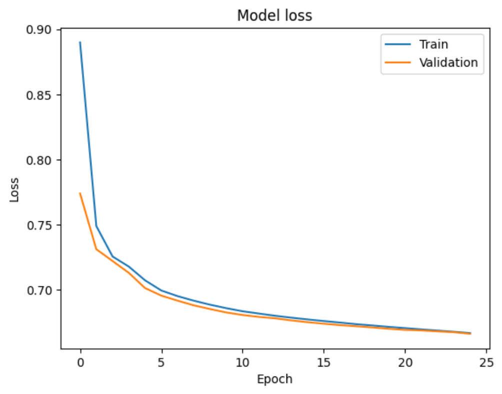
Loss curve for training session run 2.



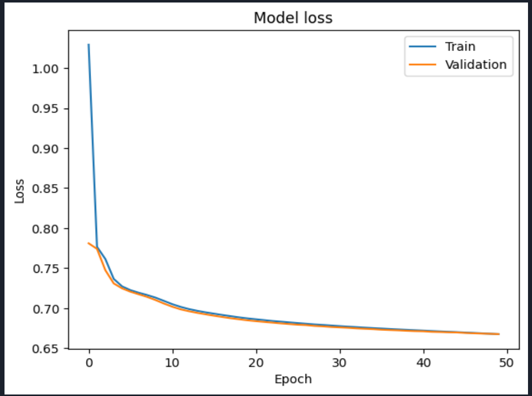
Loss curve for training session run 4.



Loss curve for training session run 10.



Loss curve for training session run 14.



Loss curve for training session run 17.

Model Performance

Training Loss (Blue Line)

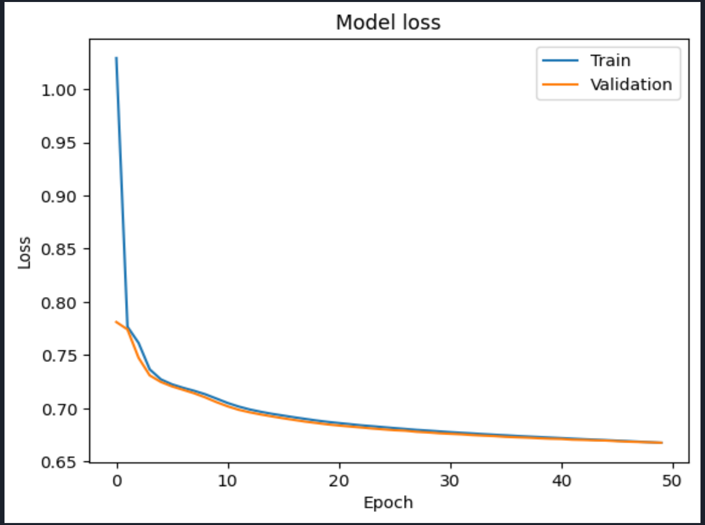
- **Initial Phase**: The training loss starts high (around 1.00) and decreases sharply within the first 10 epochs.
- **Later Phase**: After the initial drop, the training loss continues to decrease gradually and flattens out below 0.75 as it approaches epoch 50.

Validation Loss (Orange Line)

- **Initial Phase**: The validation loss also starts high and decreases sharply, similar to the training loss.
- **Divergence Point**: Around epoch 5, the validation loss starts to converge towards the training loss.
- **Later Phase**: The validation loss flattens out just below 0.75 but shows a slight increase towards the end of the epochs.

Key Observations

1. **Initial Learning**: Both training and validation losses decrease sharply, indicating the model is learning well initially.
2. **Good Fit**: The convergence between training and validation loss around epoch 40 suggests the model is learning well. Both validation and train loss have decreases to a point of relative stability. Further training would result in overfit or underfit.



LSTM-Chatbot Response Generation (non-transformer based)

Inference Model: Mimics the training model's architecture, requiring two inputs of similar shape and dimension.
Tokenization & Padding: User inputs are tokenized and padded using the same parameters as in training.
Response Generation:

- **Stepwise Prediction:** Model generates one word at a time based on the previous word in the sequence.
- **Decoding:** Predicted numbers are mapped to words using the model's vocabulary. The process continues until an 'end' token or max sequence length is reached.
- **Text Conversion:** Final tokenized response is converted to readable text using

Chatbot Response Generation (non-transformer based)

```
get_chatbot_response()

Say something Emobot would like to chat with you:  hi how are you

get_chatbot_response()

Say something Emobot would like to chat with you:  hi how are you

'you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do no
t you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not you do not'
```

BERT /GPT-2 Chatbot Architecture (transformer based)

1. **Model Overview:**
   - **BERT (Bidirectional Encoder Representations from Transformers):** A pre-trained language model achieving state-of-the-art results in NLP tasks.
   - **Task:** Fine-tuning BERT for multi-class sequence classification to classify emotions from Twitter data.
2. **Data Preparation:**
   - **Label Encoding**
   - **Tokenization**
3. **Custom Dataset Class:**
   - **EmotionDataset:** A PyTorch Dataset class for efficient data handling, providing an interface for tokenized inputs and labels, and integrating with DataLoader for batching and shuffling.
4. **Training Process:**
   - **Trainer Class:** Managed training using `transformers` library, specifying hyperparameters (epochs, batch sizes, etc.), and handled the training loop including optimization and evaluation.
5. **Model and Tokenizer Saving:**
   - **Post-Training:** Saved the fine-tuned BERT model and tokenizer for future use, ensuring efficient reuse for inference or further fine-tuning.

BERT / GPT-2 Chatbot Response Generation

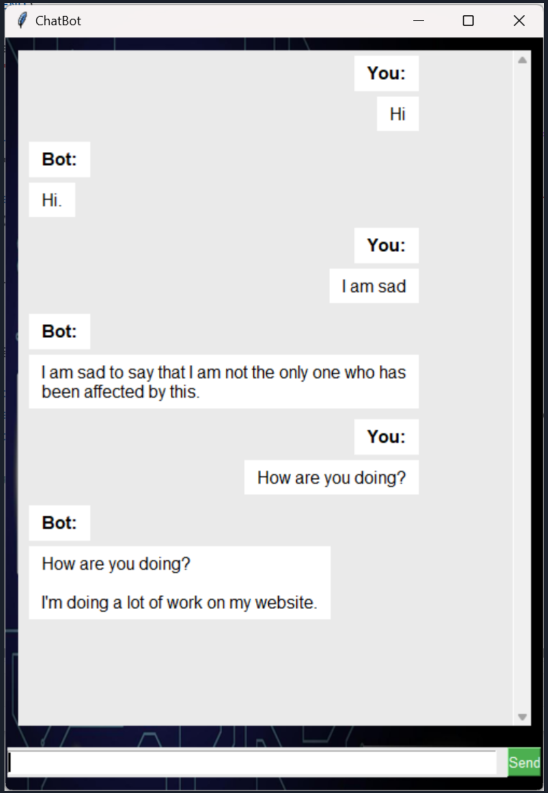
Model Initialization: Uses pre-trained GPT-2 for generating contextually relevant text.

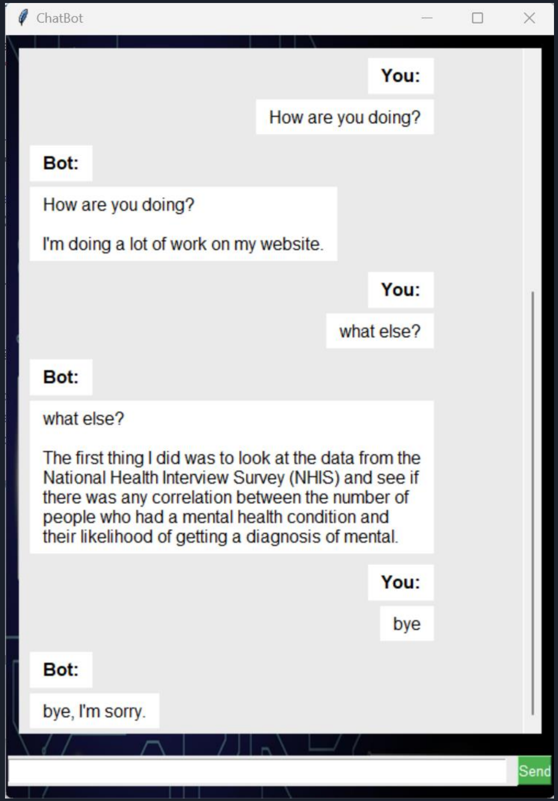
- **Response Generation:**
  - **Encoding:** User input is converted to token IDs with GPT-2 tokenizer.
  - **Text Generation:** Model generates text with specified parameters (`max_new_tokens`, `num_return_sequences`).
  - **Decoding:** Token IDs are converted back to text, omitting special tokens, and truncated at the first period.
- **Interaction:** Chatbot prompts continuously until receiving a termination command (e.g., "exit", "quit", "bye").

```
Your companion is ready! Let's chat
You:  Hi
Bot: Hi.
You:  I am sad
Bot: I am sad to say that I am not the only one who has been affected by this.
You:  How are you doing
Bot: How are you doing?

I'm doing a lot of work on my website.
You:  I am doing great
Bot: I am doing great.
You:  what else?
Bot: what else?

The first thing I did was to look at the data from the National Health Interview Survey (NHIS) and see if there was any correlation between the number of
people who had a mental health condition and their likelihood of getting a diagnosis of mental.
You:  bye
Bot: Goodbye!
```

Results

Graphic user Interface





Conclusion and Future Work

- To achieve the goal of a fully functional empathetic chatbot, there should be consideration for upgrading a hybrid model incorporating transformers like BERT and/or GPT-2/3.

- Other proposed works include allowing the model to learn from user inputs, train the model on more diverse datasets and integrate the sentiment analysis model created.

- These additional tasks should improve the model's ability to understand nuance in conversations and provide appropriate responses.

- Deploy the chatbot on a cloud platform (e.g., AWS, Google Cloud, Azure) for scalable and relia
time interaction.

References

Lakhlani, H. (2024, February 3). *AI in Everyday Life: Practical Applications.* Retrieved August 3, 2024, from https://blogpilot.ai/blogs/ai-in-everyday-life

Goddman, D. l. (2024 February 21). *Mental Health Statistics And Facts*. Retrieved August 3, 2024, from https://www.forbes.com/health/mind/mental-health-statistics/

Stephenson, E. (2023, September 22). *Insights on Canadian Society Mental disorders and access to mental health care*. Retrieved August 3, 2024, from https://www150.statcan.gc.ca/n1/pub/75-006-x/2023001/article/00011-eng.htm

Brownlee, J. (2022, September 27). *Attention in Long Short-Term Memory Recurrent Neural Networks*. Retrieved August 5, 2024, from https://machinelearningmastery.com/attention-long-short-term-memory-recurrent-neural-networks/

Brownlee , J. (2020, September 12). *Understand the Impact of Learning Rate on Neural Network Performance*. Retrieved August 6, 2024, from https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-netwo