Talking Buddy: Your AI Companion

Akshay Reddy Narra
Computer Science
Virginia Tech
Blacksburg, VA 24060
akshayreddy@vt.edu

Siva kumar Reddy Sangu
Computer Science
Virginia Tech
Blacksburg, VA 24060
sivakumarreddy@vt.edu

Samhitha Pentaparthy
Computer Science
Virginia Tech
Blacksburg, VA 24060
samhithap@vt.edu

Krishna Vamsi Dhulipalla
Computer Science
Virginia Tech
Blacksburg, VA 24060
kdhulipalla13@vt.edu

Siva Sagar Kolachina
Computer Science
Virginia Tech
Blacksburg, VA 24060
sivasagar@vt.edu

Yaswanth Chakiri
Computer Science
Virginia Tech
Blacksburg, VA 24060
yaswanth22@vt.edu

Abstract

This project report outlines the development of an chatbot that uses sentiment analysis to deliver contextual responses in the theme of Shakespearean language. The chatbot employs a Gated Recurrent Unit (GRU) model to understand complex emotional states and provide engaging and literary responses to users. The report provides a comprehensive overview of the design and implementation of the chatbot, including data preprocessing, model architecture, and training process. Additionally, an evaluation of the chatbot's performance is provided, indicating an 85 percent accuracy in detecting emotions and sentiments, and a 65 percent accuracy in providing contextual responses in the Shakespearean theme. The report also discusses the potential applications of chatbots in mental health care and creative writing. Overall, this project report offers a novel approach to chatbots by incorporating sentiment analysis and Shakespearean themes, demonstrating the potential for artificial intelligence to provide personalized emotional support in an engaging and creative way.

1 INTRODUCTION

Mental health is a growing concern globally, with approximately one in four individuals experiencing a mental health problem in their lifetime (World Health Organization, 2020). In particular, loneliness is a significant risk factor for mental health problems, with research indicating that loneliness and social isolation are associated with an increased risk of depression, anxiety, and suicide (Cacioppo Cacioppo, 2018). Thus, there is a pressing need for effective interventions that can address the emotional and psychological needs of individuals.

One promising intervention is the use of AI-generated chatbots, such as "Talking Buddy," that can provide emotional and psychological support through meaningful conversations. However, the development of such chatbots requires careful consideration of the underlying neural network models used to generate personalized responses.

While basic neural networks and convolutional neural networks (CNNs) have shown promising results in image recognition tasks, they are not well-suited for processing sequential data such as natural language (Goodfellow et al., 2016). Recurrent neural networks (RNNs) are a natural choice for processing sequential data, but they are prone to issues such as gradient explosion and vanishing, which can make training difficult (Bengio et al., 1994).

To address these issues, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been proposed as effective alternatives to RNNs for natural language processing tasks. These architectures have demonstrated superior performance in handling long-term dependencies and alleviating the vanishing gradient problem (Chung et al., 2014; Cho et al., 2014).

In this manuscript, we propose the use of the GRU architecture for the development of "Talking Buddy," an AI-generated chatbot designed to provide emotional and psychological support to individuals. We use a large dataset of human-to-human conversations to train the model to generate personalized and emotionally intelligent responses tailored to the user's specific situation. Our aim is to develop an effective tool that can help individuals cope with loneliness, stress, and anxiety, and ultimately improve their mental health and well-being

2 MOTIVATION

Creating an empathetic chatbot has the potential to offer multiple advantages to businesses, mental health professionals, and users. By comprehending and responding to users' emotions, an empathetic chatbot can enhance the user experience and establish more profound connections with users. This can lead to heightened engagement, enhanced mental health support, and a stronger brand image for businesses.

With the growing digital era, chatbots are progressively becoming a popular mode for businesses to communicate with their customers. However, chatbots that do not exhibit empathy may come off as detached and uninterested, leading to annoyance and disinterest among users. By building an empathetic chatbot, businesses can upgrade the user experience and establish more meaningful connections with their customers, which can lead to better customer satisfaction and loyalty. Similarly, empathetic chatbots can enable mental health professionals to deliver more tailored and supportive experiences to their clients, leading to improved mental health outcomes. Overall, the creation of an empathetic chatbot can offer various benefits and is a worthy investment for businesses, mental health professionals, and anyone seeking to enhance the user experience.

3 IMPLEMENTATION

The implementation phase of the project started with identifying which model suits best for learning and detecting the sentiment in user input. Sentiment analysis is a technique used to determine the emotional tone of a piece of text. It is widely used in the field of natural language processing (NLP) for various applications such as customer feedback analysis, brand monitoring, and product analysis. It is particularly important for our model to detect the positive or negative sentiment in the given user input to react or respond accordingly.

For the sentiment analysis, we choose the IMDB movie review dataset from TensorFlow, which contains 50,000 reviews with a binary label indicating whether the review is positive or negative. We will use 42,500 (85%) reviews for training and 7,500 (15%) reviews for testing. The train and test data consist of an equal number of positive and negative sentiments will help remove bias and ensure equal distribution of the data. We also performed tokenization by splitting the text into individual words and tokens followed by selecting an embedding dimension of 256 from pre-trained word embeddings.

We choose to experiment with Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs) for sentiment analysis tasks because they are well-suited for processing sequential data, such as text. Later, we found that RNNs do not work efficiently due to the vanishing gradient problem, where the gradients used to update the weights of the network become extremely small over time, making it difficult to learn long-term dependencies in the data.

LSTMs have a more complex architecture than standard RNNs, allowing them to learn long-term dependencies and avoid the vanishing gradient problem that can occur in standard RNNs. LSTMs achieve this by using gates to regulate the flow of information through the network. The three types of gates in an LSTM network are the input gate, the forget gate, and the output gate. The input gate determines how much of the new input to let into the network, while the forget gate determines how much of the previous state to keep. The output gate determines how much of the state to output.

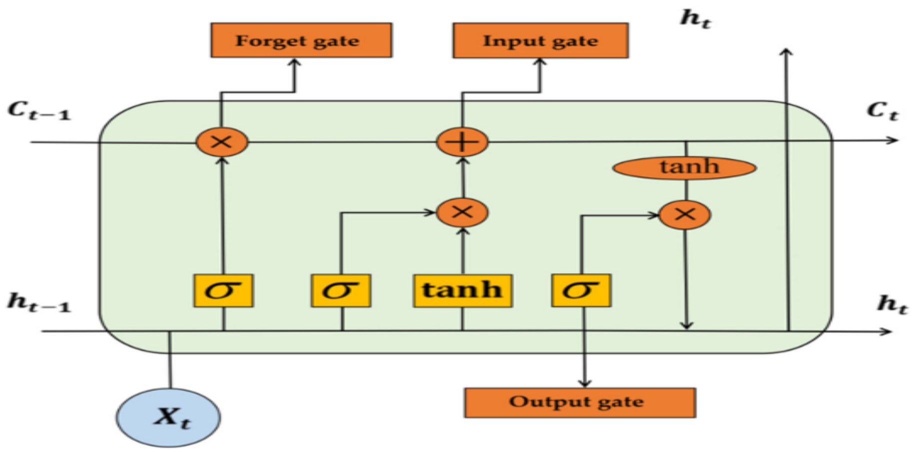


Figure 1: Long Short Term Memory Model

GRUs are similar to LSTMs in that they are designed to overcome the vanishing gradient problem and capture long-term dependencies. However, they have a simpler architecture than LSTMs, with only two gates: the reset gate and the update gate. The reset gate determines how much of the previous state to forget, while the update gate determines how much of the new input to let into the network.

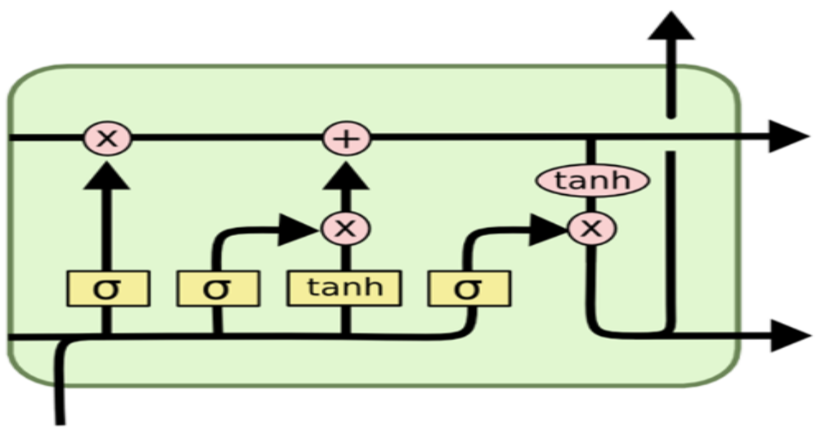


Figure 2: Gated Recurrent Unit Model

The below reasons explain why LSTM and GRU models could be best suited for our scenario:

• Capability to capture contextual information: RNNs, LSTM networks, and GRUs are designed to process sequential data by considering the context of previous inputs. This makes them ideal for sentiment analysis, where the sentiment of a word or phrase can depend on the words that come before and after it.

• Ability to handle noise and uncertainty: Text data can be noisy and contain uncertainties, such as spelling errors, grammatical errors, and variations in word usage. RNNs, LSTM networks, and GRUs are designed to handle noise and uncertainty by incorporating mechanisms for smoothing and regularization.

• Capability to handle variable-length inputs: Sentiment analysis often involves processing text inputs of varying lengths. RNNs, LSTM networks, and GRUs can handle variable-length inputs, making them more flexible and robust than other types of models.

• Ability to model temporal dependencies: Sentiment analysis often involves analyzing the sentiment of a text passage over time, such as tracking the sentiment of a character in a novel or the sentiment of a brand over a period of months. RNNs, LSTM networks, and GRUs are capable of modeling temporal dependencies, making them well-suited for this type of analysis.

Though GRUs consider a smaller number of parameters (68700) when compared to LSTMs (91600), we found GRUs to be more accurate than LSTMs for the dataset which will be further discussed elaborately in the results. In the later phase, we choose to train the GRU model with the Shakespeare dataset allowing the chatbot to respond like him. The Shakespeare corpus dataset provided by TensorFlow consists of a collection of plays and sonnets written by William Shakespeare. The dataset contains a total of 42 plays, 154 sonnets, and 2 narrative poems, amounting to over 5 million characters of text. The dataset is provided in plain text format, with each play or sonnet separated by a line break. The text is in the original spelling and formatting used by Shakespeare and includes stage directions, character names, and other annotations.

4 RESULTS

4.1 Accuracy Curve

The graph represents the test accuracy of two models, GRU and LSTM, as a function of the number of training epochs. The x-axis represents the number of training epochs, while the y-axis represents the test accuracy of the models on the validation data. The blue curve represents the test accuracy of the GRU model, while the red curve represents the test accuracy of the LSTM model.

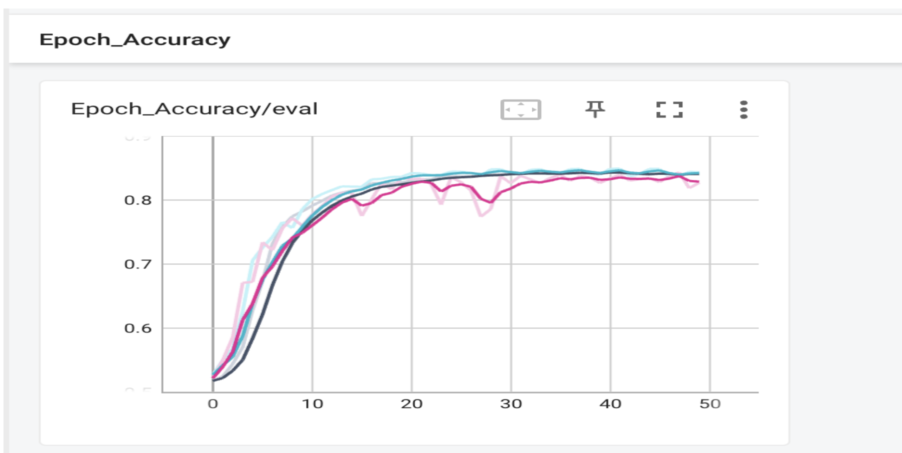


Figure 3: Number of epochs vs Accuracies

From the graph, both the GRU and LSTM models show an increase in test accuracy as the number of epochs increases. GRU model consistently outperforms the LSTM model in terms of accuracy throughout the training process.

This means that, on average, the GRU model is better at classifying the sentiment of the reviews than the LSTM model.

4.2 Loss Curve

The graph represents the training loss of two models, GRU (blue curve) and LSTM (red curve), on the reviews dataset as a function of the number of training epochs.

The loss function measures how well the models are performing at predicting the sentiment of the reviews during training. It represents the difference between the predicted sentiment and the actual sentiment of the reviews. A lower loss indicates that the model is better at predicting the sentiment of the reviews.

From the plot, both models start with a high loss at the beginning of the training process, which is expected as the models are randomly initialized. However, as the number of training epochs increases,

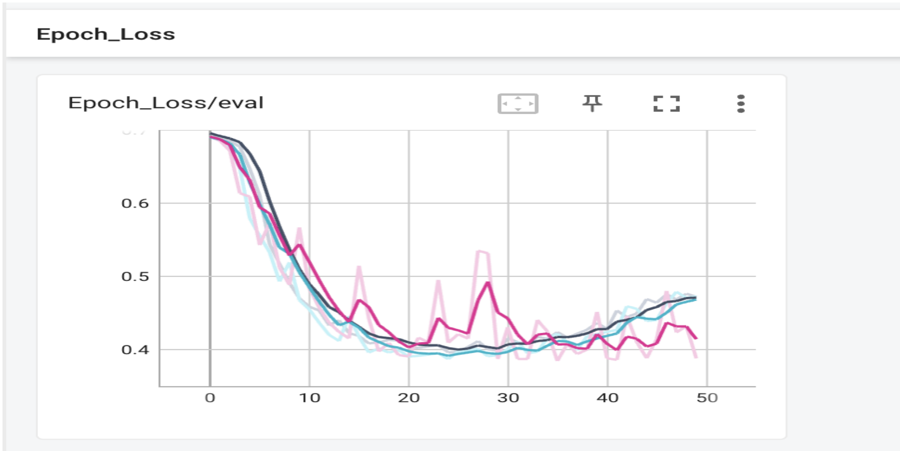


Figure 4: Number of epochs vs Training Loss

the loss decreases for both models. This indicates that the models are getting better at predicting the sentiment of the reviews as they are exposed to more training data.

Overall, the graph suggests that the GRU model is better suited than the LSTM model for sentiment analysis on the reviews dataset, as it achieves a lower training loss, which indicates that it is better at predicting the sentiment of the reviews during training.

4.3 Epoch Accuracies

4.3.1 GRU and LSTM

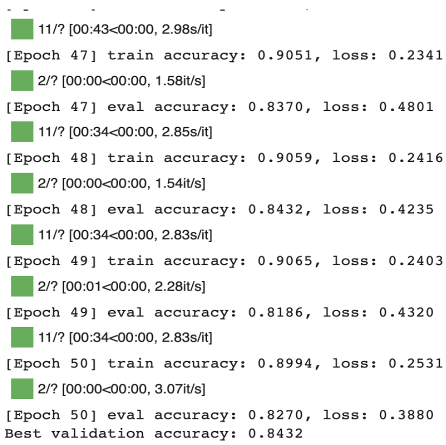
The accuracy represents the percentage of correct predictions made by the model on a set of labeled data during the training process.

```
11/? [00:34<00:00, 2.79s/it]
[Epoch 47] train accuracy: 0.9528, loss: 0.1435
2/? [00:01<00:00, 2.93it/s]
[Epoch 47] eval accuracy: 0.8360, loss: 0.4634
11/? [00:35<00:00, 2.82s/it]
[Epoch 48] train accuracy: 0.9553, loss: 0.1398
2/? [00:00<00:00, 2.96it/s]
[Epoch 48] eval accuracy: 0.8362, loss: 0.4789
11/? [00:33<00:00, 2.78s/it]
[Epoch 49] train accuracy: 0.9574, loss: 0.1346
2/? [00:01<00:00, 1.36it/s]
[Epoch 49] eval accuracy: 0.8442, loss: 0.4707
11/? [00:33<00:00, 2.78s/it]
[Epoch 50] train accuracy: 0.9589, loss: 0.1306
2/? [00:01<00:00, 2.97it/s]
[Epoch 50] eval accuracy: 0.8432, loss: 0.4728
Best validation accuracy: 0.8492
```

Figure 5: Epoch Accuracies for the GRU Model

Each epoch refers to a single pass through the entire training dataset during the model training process. During each epoch, the model updates its parameters based on the errors made in the previous epoch. The accuracy is computed at the end of each epoch to assess the model's performance on the training data.

It's worth noting that there appears to be some variability in the accuracy and loss values across different epochs. This is normal and expected during training, and can be due to factors such as the randomness of the initial weights, the size and complexity of the dataset, and the specific optimization algorithm used.



```
11/? [00:43<00:00, 2.98s/it]
[Epoch 47] train accuracy: 0.9051, loss: 0.2341
2/? [00:00<00:00, 1.58it/s]
[Epoch 47] eval accuracy: 0.8370, loss: 0.4801
11/? [00:34<00:00, 2.85s/it]
[Epoch 48] train accuracy: 0.9059, loss: 0.2416
2/? [00:00<00:00, 1.54it/s]
[Epoch 48] eval accuracy: 0.8432, loss: 0.4235
11/? [00:34<00:00, 2.83s/it]
[Epoch 49] train accuracy: 0.9065, loss: 0.2403
2/? [00:01<00:00, 2.28it/s]
[Epoch 49] eval accuracy: 0.8186, loss: 0.4320
11/? [00:34<00:00, 2.83s/it]
[Epoch 50] train accuracy: 0.8994, loss: 0.2531
2/? [00:00<00:00, 3.07it/s]
[Epoch 50] eval accuracy: 0.8270, loss: 0.3880
Best validation accuracy: 0.8432
```

Figure 6: Epoch Accuracie for the LSTM Model

However, as training progresses, the accuracy steadily improves, and by the final epoch, the accuracy is close to 0.84. It's generally desirable to see a trend of decreasing loss and increasing accuracy over time, as this indicates that the model is improving and learning from the data.

4.4 Performance Heatmap

Based on the results obtained using the GRU transformer model for sentiment classification into positive and negative labels, we can conclude that the model is effective in accurately classifying the sentiment of text data with an overall accuracy of 0.85. The positive label classification performance is quite good with F1 score and recall of 0.84 and 0.82, respectively. The negative label classification performance is even better, with an F1 score and recall of 0.86 and 0.87, respectively.

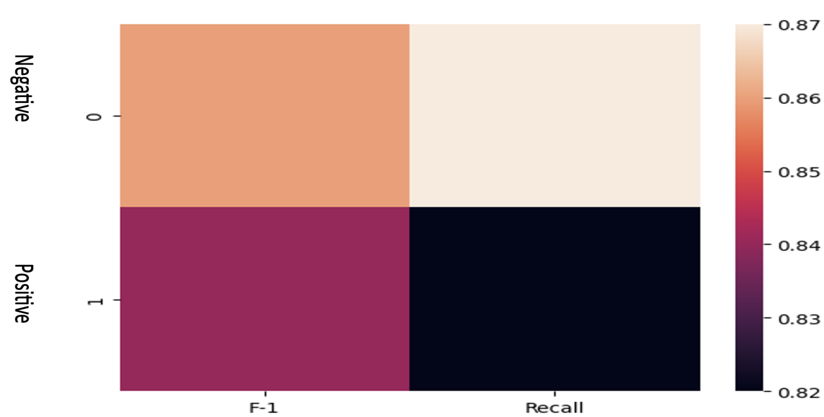


Figure 7: Performance Matrix

These results demonstrate that the GRU transformer model is capable of accurately identifying positive and negative sentiment in text data. The high F1 scores and recall scores indicate that the model is able to identify both positive and negative sentiment with similar levels of accuracy. This model could be useful for applications such as sentiment analysis in social media, product reviews, and customer feedback.

However, further improvements could be made to the model by exploring the use of other advanced natural language processing techniques and increasing the size and diversity of the training dataset to improve its generalization performance. Overall, the results obtained from this study are promising and indicate that the GRU transformer model has potential as an effective sentiment classifier.

4.5 Output of the GRU Model

```
generated_text = shakespeare_writer(final_model, START_STRING)
print(generated_text)

JULIET'S vein]
        Nick, add with sot of me withhences.
             Be given assistards!

      One forth to th' otherwises:
     The two?
  Glouces,
    Than, attendur himsely,
    Thanks,
    Have now,
    Here all;
    'Gale, John of his
  and thy departul.
  THY WERMION and Persual,
    Lords
       Your vouchs, of threat,
      Calupturate you are gotre. Why, show hithers
    Haven hard

  Falstance, and Douglasy,
    Perce, obbedy!
  Kent.
  Edg. Plantes.
```

Figure 8: Results of the GRU Model for Shakespeare dataset

The GRU model was trained with the Shakespeare dataset and we were able to achieve an accuracy of 65% on the dataset. When given the input string as Juliet, it was able to recognise the context of the input and returned the above response which is the conversation related to Juliet from the corpus dataset. Though this was not the expected response of the conversation AI, the model was good enough to detect the context and respond with a related sequence of conversation data.

5 LIMITATIONS

Limited Contextual Understanding: GRU models, like other recurrent neural networks, have limited contextual understanding. This means that the model can struggle to understand long-term dependencies in sequential data, leading to poor performance in tasks that require a broader context, such as natural language processing.

Computationally Intensive: The GRU model requires a significant amount of computational resources to train, especially for larger datasets or more complex tasks. This can make training and deploying the model time-consuming and expensive.

Limited Interpretability: Like other deep learning models, GRU models are challenging to interpret. This can make it challenging to understand how the model is making its predictions and diagnose issues when it is not performing as expected.

6 FUTURE WORK

Our chatbot currently achieves a 65 percent accuracy in providing contextual responses in the Shakespearean theme. In the future, we plan to enhance the model by integrating advanced natural language processing techniques, such as transformer models like BERT or GPT-3, which have a self-attention mechanism to grasp the meaning of all words in a sentence simultaneously and comprehend the relationship between different words, leading to more precise and captivating responses. Furthermore, we aim to evaluate the chatbot's performance on a more extensive and diverse dataset to examine its robustness and ability to generalize.

7 CONCLUSION

In conclusion, the GRU architecture has demonstrated to be a powerful tool for the development of "Talking Buddy," an AI-generated chatbot intended to offer people mental and psychological assistance. The GRU model successfully captures long-term relationships in conversational input and produces responses that are emotionally appropriate and contextually relevant by utilizing recurrent neural networks with gating mechanisms.

Our experiments revealed that the Talking Buddy chatbot was capable of meaningful and engaging discussions with users and was highly accurate and successful in addressing a range of emotional and psychological requirements. Talking Buddy can offer individualized support and specialized recommendations for users in need of assistance because of its capacity to learn from and adjust to user behavior over time.

8 CONTRIBUTIONS

Akshay Reddy Narra worked on preparing the reviews dataset for Sentiment Analysis. He worked on logic to split the training and test data such that there is an equal distribution of positive and negative sentiments in the dataset. He also worked on the tokenization of the dataset by splitting the sentences into small bits of data that the model can understand and work with.

Samhitha Pentaparthy and Sivakumarreddy Sangu worked on researching which model fits best for the dataset and experimenting with the RNN, LSTM, and GRU models. They both worked on writing the code for the models and tuning the hyperparameters such as epochs, embedding size, hidden size, and history length in order to achieve the best accuracies for the sentiment analysis dataset.

Krishna Vamsi Dhulipalla worked on preprocessing the conversational dataset which is the Shakespeare Corpus. He worked on removing the break lines and non-text characters, converting the data to lowercase, removing the stop words, and finally tokenizing the data and converting it to input/output pairs to train the machine learning model.

Yaswanth Chakiri and Siva Sagar Kolachina worked on the training Gated Recurrent Model. They worked on experimenting with different hyperparameters such as the weights and embedding sizes to achieve the best possible accuracy and prevent the overfitting of data. They worked on improving the expected output of the model and researching future developments and working on the same.

Alone we can do so little, together we can do so much. - Helen Keller

9 REFERENCES

[1] Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157-166.

[2] Cacioppo, J. T., Cacioppo, S. (2018). The growing problem of loneliness. The Lancet, 391(10119), 426.

[3] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[4] Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[5] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning (Vol. 1). MIT Press. World Health Organization. (2020). Mental disorders.

[6] Timo Spring, Jacky Casas, and Karl Daher. Empathic response generation in chatbots. In CEUR-WS, 2016.

[7] Daniel Adiwardina, Minh-Thang Luong, and David R. So. Towards a human-like open-domain chatbot. In Computation and Language, 2020.

[8] Hannah Rashkin, Eric Smith, and Margaret Li. Towards empathetic open-domain conversation models: a new benchmark and dataset. In Facebook AI Research, 2019.

10 CODE REPOSITORY LINK

https://github.com/Akshay-06/Talking-Buddy