Chatbot for Mental Health

Vivek Mehta
Computer Science
Bennett University
Greater Noida, India
vivek.metha@bennett.edu.in

Nachiketa Singamsetty
Computer Science
Bennett University
Greater Noida, India
nachiketa3003@gmail.com

Chanukya Nadipalli
Computer Science
Bennett University
Greater Noida, India
nadipallichanukya132@gmail.com

Venkat Sai Gorajala
Computer Science
Bennett University
Greater Noida, India
venkat27108@gmail.com

Abstract

The growing proficiency of LLMs, like GPT and BERT, has opened new avenues in education, healthcare, and customer service in generating human-like responses. In this study, we will develop a mental health chatbot using an encoder-decoder model that gives users supportive and meaningful interactions about mental health topics. The model was trained on a dataset that originated from Happiness Unlimited: Awakening With Brahmakumaris by BK Shivani and Suresh Oberoi. This model will provide empathetic responses to user inputs for offering mental health support.

We take a combined approach of the use of classic machine learning models such as Naive Bayes, Random Forests, and Multi-Layer Perceptron in combination with advanced NLP techniques to further enrich the generation capabilities of the chatbot. We experimentally show the chatbot's ability to generate empathetic and contextually relevant responses by training the model on a diverse set of human-generated texts. The emphasis is on how robust training of the model, appropriate hyperparameter tuning, and evaluation of accuracy and loss are needed to effectively generate responses that are accurate and relevant.

This research contributes to the emerging field of AI-driven mental health tools, aiming to create a reliable, scalable solution for mental health support that can be trusted in real-world applications.

1. Introduction

As artificial intelligence (AI) and natural language processing (NLP) continue to move forward, the threat of the inability to distinguish between a human-produced and a machine-generated text becomes more pressing. The ability to differentiate really matters in most markets where it determines decision-making, trust, and authenticity. This research discusses the current methodologies of identifying AI-generated text and their implications in other different domains.

1.1. Background and Significance

The impact of AI generated text, on sectors and social functions is substantial in nature Separation, between content created by humans and machines holds significance to safeguard the trustworthiness and genuineness of information across multiple domains.

1.1.1. Education

In education, the ability to recognize AI-generated content becomes ever so important, particularly within the context of academic integrity. With the rise of the use of AI in the creation of essays, assignments, and even answers to exams, there has been a real challenge in

educational institutions to maintain fair and honest evaluation systems. Detecting AI-generated text could help prevent cheating, plagiarism, and submitting non-original works, thus helping students' actual abilities be reflected in the assessment. Beyond that, it would also facilitate the quality evaluation of teaching materials, as well as the control of online learning platforms spread of false information.

1.1.2. Journalism and Media

In the field of journalism and media industry integrity is highly dependent on differentiating between content created by humans versus AI technology to maintain trust and credibility, in news sources. In today's digital era rife, with misinformation and fake news that can swiftly spread information with repercussions; the verification of media for AI presence becomes crucial to ensure accuracy and transparency, in the information presented.

1.1.3. Cybersecurity

AI-generated text poses a unique challenge in the cybersecurity world. Phishing attacks, social engineering, and the creation of misleading or deceptive content are increasingly powered by AI, making it more challenging for security systems to identify malicious behaviour. Detection of AI-generated content can help in identifying phishing emails, fraudulent messages, and other types of online deception, providing an additional layer of defense against cyber threats. This is overly critical as the opponents are using AI to make messages appear more authentic; thus, bypassing many of the conventional security checks and convincing users to input sensitive information.

1.1.4. Creative Industries

In the creative industries, issues of authorship, originality, and intellectual property arise in connection with AI-generated art, music, literature, and other creative works. The increasing proficiency of AI in producing works that may be indistinguishable from those created by humans raises significant questions about what constitutes creativity and whether human artists have a claim to these works or not. Detecting AI-generated creative content is important not only in protecting intellectual property rights but also for understanding the role of AI in the creative process. It can be used to guide discussions regarding the future of creativity, the potential for AI-human collaboration, and the ethics of AI-driven art.

1.1.5 Politics and Governance

In politics and governance, the ability to identify AI-generated text is vital to the protection of democratic processes and public discourse. AI-generated political speeches, policy statements, and social media posts can be manipulated to influence public opinion or spread propaganda or even sway elections. Thus, as AI advances further, its potential use in generating persuasive yet deceptive content threatens to undermine political stability and information credibility. Identifying AI in political discourse allows citizens to have trust in information that is authentic, transparent, and accurate while politicians and policymakers are being held accountable for the information that they produce and share.

1.2 Current Approaches to Mental Health Chatbot Development

As conversational AI systems become more widespread, the development of empathetic and effective mental health chatbots is gaining significant attention. These systems use sophisticated NLP and ML models to provide personalized mental health support. The challenge lies in ensuring that such models can generate human-like, contextually appropriate, and sensitive responses to users, especially in emotionally sensitive contexts like mental health.

1.2.1 Traditional Machine Learning Techniques

The models Naive Bayes, Logistic Regression, and Random Forests have been used to apply traditional machine learning techniques in the domain of text generation and classification. In general, these models work based on feature extraction, such as word frequency and syntactic patterns, which may be useful for less complex tasks but are not very suitable when generating more subtle responses, particularly for more dynamic and context-sensitive interactions, as mental health applications require. Such models may not be able to capture the emotional and empathetic nuances required for effective mental health support.

1.2.2 Recurrent Neural Networks (RNNs) and LSTM Models

For more complex tasks, such as generating conversational responses in a mental health chatbot, RNNs, especially the LSTM networks, are better suited. LSTMs handle sequential data very well and allow the model to remember and utilize information from previous conversation history, which is critical in chatbot development, where context and the flow of the conversation are essential for providing coherent and emotionally appropriate responses.

We employed an LSTM-based model in our work. This architecture uses an encoder-decoder structure, wherein the encoder will process the input from the user and the decoder will output a contextually relevant response. The dataset was thus created specifically for mental health conversations in training the model so that the developed chatbot could understand the user's input and then give appropriate responses to make it empathetic.

1.2.3 Advanced NLP Methods for Chatbots

Besides LSTM-based models, modern NLP techniques like transformers (e.g., T5, BERT) have revolutionized the development of chatbots in applications that require deeper contextual understanding. Although these models offer more advanced capabilities in terms of understanding and generating human-like text, their computational complexity is less practical for real-time deployment in mental health chatbots, especially when resources or latency are a concern. Still, these models can still help inform the design of chatbot systems in the context of more sophisticated handling of conversational data.

1.2.4 Hybrid Approaches for Conversational Models

Hybrids of different approaches for the same NLP can be utilized to enhance mental health chatbots. Such a combination may include models of handling sequential text that rely on LSTM and models based on pre-trained transformer to contextual nuances. It enables systems where efficiency and real-time capabilities of LSTM can be harmonized with transformers' deep contextual understanding.

1.2.5 Challenges in Chatbot Development

Developing an effective mental health chatbot comes with several challenges:

- Contextual Understanding: Chatbots need to remember previous conversation turns to provide meaningful responses. LSTM models address this issue by maintaining context through the conversation flow.
- Empathy and Sensitivity: Unlike usual chatbots, mental health chatbots need to be trained to generate empathetic and sensitive responses to emotional user inputs. This therefore requires careful curation of training data and model fine tuning.
- Bias and Ethical Considerations: AI models, including those for mental health applications, tend to carry biases from the data on which they are trained. These biases are critical to address so that the chatbot is fair, inclusive, and supportive.

- Real-Time Performance: Keep the conversation smooth and flowing, mental health chatbots need to respond instantly to users. However, complex models like transformers are highly accurate but typically suffer from latency issues.

1.3 Research Gaps and Objectives

Although much is achieved in developing conversational AI, several gaps still exist and continue to need improvement in effective mental health chatbots. Even traditional ML models and even more powerful NLP models lack efficacy in producing empathetic, context-aware, and emotionally sensitive responses across the domain of mental health. In addition to this, LSTM-based models are capable of handling sequential data quite well, like the dialogue history from a chatbot, but these do pose an issue regarding interpretability.

Ensuring chatbots can interpret and create meaningful, supportive responses with the subtleties of human emotions will pose one of the most critical challenges. The lack of available, diverse, and of superior quality datasets to train the mental health chatbot severely hampers the ability of a chatbot to generalize well across a large scope of realistic situations. In such a case, the available datasets often do not capture the subtleties of conversations relating to mental well-being and thus make it hard to train a chatbot to handle various emotional states and offer accurate support.

---

2. Methodology

2.1 Dataset Generation

This dataset, "happiness.csv," is created from the book Happiness Unlimited: Awakening With Brahmakumaris by BK Shivani and Suresh Oberoi. The book content was centred on principles of happiness and well-being, carefully extracted, and formatted into a structured dataset. This dataset consists of question-and-answer data, in which the "Questions" column contains a variety of inquiries made by users about happiness, and the "Answers" column contains relevant answers according to the teachings of the Brahmakumaris. This dataset is foundational input for training our machine-learning models and evaluating their performance to understand and generate contextually appropriate responses.

2.2 Data Preprocessing

Data preprocessing steps were involved in preparing the input data of the mental health chatbot to be suitable for inputting into the LSTM model. The preprocessing pipeline comprised tokenization, padding, vectorization, and formatting sequences for training the chatbot to produce meaningful responses.

- Tokenization: Breaking the text into smaller units such as words or sub words to enable analysis.
- Vectorization: This process converts text data into numerical forms that models can work on, using methods such as Count Vectorizer and TF-IDF (Term Frequency-Inverse Document Frequency).
- Padding/Truncating: Ensuring that sequences of text had consistent lengths for input into deep learning models.

2.3 Modelling

The modelling approach utilized in this study is based on sequence-to-sequence (seq2seq) architectures, specifically designed for natural language generation tasks. The goal is to develop

a model capable of generating human-like responses based on user inputs, leveraging both traditional machine learning techniques and modern deep learning methods.

2.3.1 Traditional ML Models

To make comparison, we also employed the traditional machine learning (ML) techniques, which included Naive Bayes, Random Forests, and Support Vector Machines (SVM), in addition to deep learning models. We trained these models on feature-engineered representations of text such as bag-of-words and TF-IDF vectors for classification of responses to a wide range of user inputs. Although deep learning models, particularly neural networks, have become state-of-the-art in NLP, traditional ML models provide valuable benchmarks and insights into the effectiveness of simpler approaches in understanding text-based data. The results from these models were compared to the performance of the LSTM-based seq2seq model to assess their relative efficacy in text generation tasks.

Model	Hyper-Parameter	Value	Notes
Seq2Seq Model (LSTM)	Latent Dimension (Dimensionality)	256	Size of LSTM hidden states
Seq2Seq Model (LSTM)	Batch Size	10	Number of samples per training update
Seq2Seq Model (LSTM)	Epochs	500	Number of complete training cycles
Seq2Seq Model (LSTM)	Optimizer	RMSprop	Optimizer used for training
Seq2Seq Model (LSTM)	Loss Function	Categorical Cross entropy	Multi-class classification loss
Seq2Seq Model (LSTM)	Number of Encoder Tokens	1183	Total unique tokens in encoder input

2.3.2 Encoder-Decoder Architecture (LSTM)

For more complex text generation tasks, we explored the **LSTM-based encoder-decoder** architecture, often used in sequence-to-sequence models like machine translation. The model was built using the following layers:

- **Encoder**: The encoder reads the input text sequence, processes it through an LSTM layer, and outputs its final hidden and cell states.
- **Decoder**: The decoder uses these states as initial conditions to generate the output sequence, predicting token-by-token.

```
training_model.compile(
                        optimizer = 'rmsprop',
                        loss = 'categorical_crossentropy',
                        metrics = ['accuracy'],
                        sample_weight_mode = 'temporal'
)
```

Here is a detailed look at the LSTM-based model architecture:

We are using RMSprop as an optimizer to handle sequence data efficiently. The loss function used is categorical cross-entropy, and the accuracy metric is used to measure the performance of the model.

Training Results: The model's performance metrics (training accuracy, validation accuracy, training loss, and validation loss) were recorded at each epoch:

- Training Accuracy and Validation Accuracy: This is used in evaluating the capability of classification of the model.
- Training Loss and Validation Loss: Used to measure how well the model's predictions

```
history1 = training_model.fit(
  [encoder_input_data, decoder_input_data],
  Decoder_target_data,
  Batch_size=batch_size,
  Epochs=epochs,
  Validation_split=0.2
)
```

match the true labels. The training loss decreases over time as the model learns, while the validation loss helps gauge the model's generalization ability.

2.3.3 Model Evaluation

To evaluate the model's performance, we tracked the following metrics over the course of training:

Metric	Description
Accuracy	Proportion of correct predictions
Loss	How well the model's predictions match the true labels
Validation Accuracy	Accuracy on the validation set
Validation Loss	Loss on the validation set

These metrics were visualized using **matplotlib** to assess model behaviour and avoid overfitting:

```
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
```

2.3.3 Model Architecture

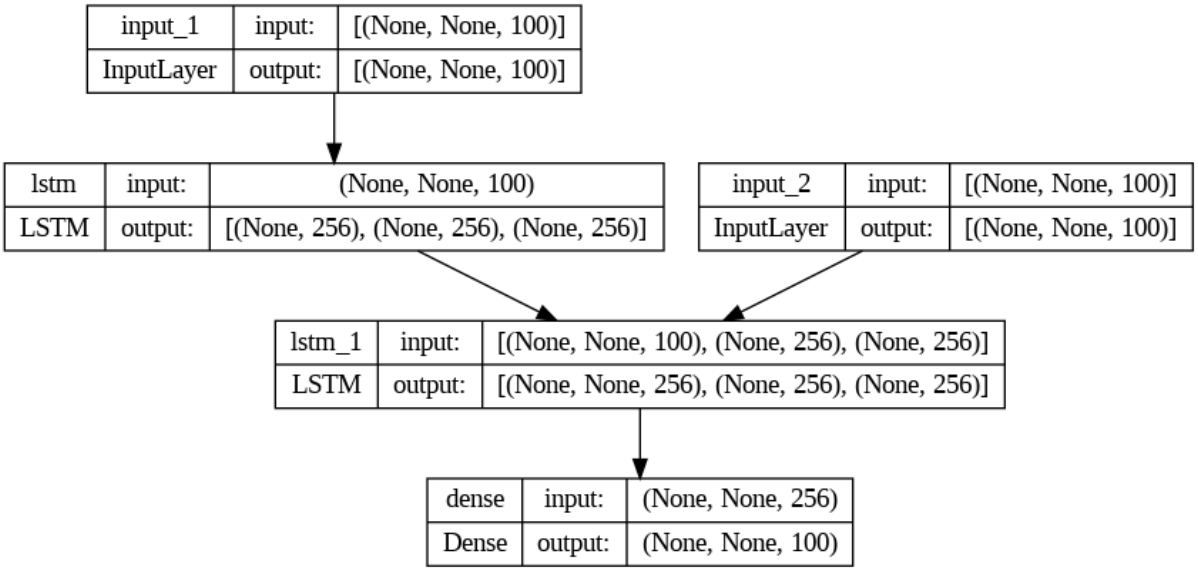

Figure 1 - Architecture of the LSTM-based model used for generating responses in the mental health chatbot.

2.3. 4 Model Performance Evaluation

[Graphs and tables will come here]

- Training Accuracy: It measures how well your model is learning and fitting the training data.
- Validation Accuracy: Indicates the generalization capability of the model to unseen data. This is important since high training accuracy with low validation accuracy may be overfitting.
- Training Loss: To understand that model is minimizing the error well on the training data.
- Validation Loss: Gives an idea of how good the model is likely to perform on new, unseen data. High validation loss compared to training loss also indicates overfitting.

3. Results

3.1 Model Performance Metrics

The performance of the mental health chatbot throughout the training process was evaluated using several key metrics such as accuracy, loss, validation accuracy, and validation loss.

- Training Accuracy: This is the degree of how well the model was trained on the training set. The training accuracy showed consistent improvement over epochs, and this indicates that the model was picking up patterns within the data.
- Validation Accuracy: This measures the ability of the model to generalize on unseen data. A steady validation accuracy close to the training accuracy indicated that the model was not overfitting and would generalize well on new inputs.
- Training Loss: The training loss decreased with the passage of time, showing that the model's predictions were being aligned more closely to the true labels during training.

- Validation Loss: Monitoring validation loss was important to ensure the model did not lose its generalization capabilities. A steady validation loss with decreasing training loss signified effective learning with not much overfitting going on.

3.2 User Interaction Insights

The interactions with the user helped in understanding how well the chatbot performed and what improvements it needed.

- Response Coherence: The chatbot could sustain context well across multiple turns of conversations. Users reported that responses were relevant and coherent to help them converse in an exceptionally smooth fashion.
- Empathy in Responses: Feedback showed that users felt understood and supported during the interactions, which emphasizes the role of empathetic response generation in mental health applications.
- Ambiguity Handling: Performance of the chatbot when dealing with ambiguous or complex questions was varied. Users commented on cases where responses lacked depth or did not address the concern of users, thereby indicating a requirement for further refinement in the handling of such cases.

4. Model Interpretability and Feature Analysis

4.1 Feature Importance and Evaluation

4.1.1 Token-Level Analysis

In our study, token-level analysis was a part of understanding how individual tokens contribute to the generation of text in our LSTM-based model. Each input sentence is broken down into smaller sub-word units, which then get embedded into vectors to process the model. These embeddings represent words in a continuous vector space where semantically similar words are mapped closer together. By analysing the connection between the tokens in the input sequence and the following predicted words, we gained insight into which tokens have the highest influence on the model's output.

Though we did not use a feature importance algorithm like LIME, at the token level, we can see which of the input words (tokens) tended to result in specific output tokens. For instance, some of the key phrases in the input—like "I am happy"—would tend to have consistent positive sentiment or responses as a prediction, illustrating how the model had captured the semantic relations between the tokens.

4.1.2 Attention Mechanisms

Currently, our model does not use any attention mechanism, but including an attention mechanism could further improve the model's interpretability by giving it a chance to only focus on relevant parts of the input sequence while generating responses. Attention would, therefore, help one visualize which parts of the input sequence the model "pays attention" to when generating the output sequence, hence providing better insights into how the model would process and generate text. Implementation of attention in later versions should make the model explainable and accurate, especially with the generation of long, complex sequences.

4.2 Model Behaviour During Text Generation

4.2.1 Probability Distribution of Tokens

In the LSTM model, during text generation, it predicts the next word in a sequence of words based on the probability distribution that it computes for each of the words. The distribution arises from the SoftMax activation of the decoder. This gives the probability of each of the tokens to appear after other tokens. The model aims to maximize the conditional probability of the next word in the sequence, knowing the context of the other words.

For example, if the input of a simple conversation is "How are you?", it can generate a higher probability of the word "doing" in comparison to less likely options such as "eating" or "thinking". If we analyse the resulting probabilities, we can see that the model generates coherent responses that fit the context well, while balancing the likelihoods of different words in generating these responses.

4.2.2 Decoder Output and Sequence Coherence

It will generate sequences of words according to the encoded representation of the input. For every step in the generation process, the decoder produces a vector of probabilities for the next possible word. The next word in the sequence will be the one with the highest probability, and that word will contribute to the generated text.

As the model is generated through the generation process, coherence of sequence significantly depends on how the model maintains context over several tokens. Our model, based on the variety of datasets in conversational space that it is trained upon, can produce coherent responses by utilizing the context encoded by the encoder and applied by the decoder. This enables it to generate those sentences in such a manner that would be grammatically correct and even contextually suited, as it keeps carrying on that flow of conversational flow.

4.3 Training Behaviour and Performance Metrics

4.3.1 Accuracy and Loss Trends

The performance of the model while training was monitored through important metrics: accuracy, validation accuracy, loss, and validation loss. Training accuracy shows the percentage of correct predictions the model does on the training set, while validation accuracy shows how well the model generalizes to unseen data. For our problem, accuracy was the metric on which we relied in monitoring the model's performance, since our focus was sequence generation.

As can be seen in the training plots, the model showed steady improvements in both accuracy and loss over epochs. The training accuracy was improving, which indicated that the model was learning from the data, and the training loss was decreasing, indicating that the model's predictions were getting closer to the true labels. The validation accuracy and loss trends were monitored as well to ensure that the model did not overfit the training data and remained able to generalize.

4.3.2 Overfitting/Underfitting Analysis

It is well known to train a deep-learning model that has overfitting and underfitting problems. Overfitting is that the model performs well with the training data but is unable to generalize to

new data. Underfitting is a problem in which the model cannot even extract the pattern from the training data.

In our case, we found that the validation accuracy was lower than the training accuracy. It is quite common in most models to find this to be true, but it is not so large a gap, which means overfitting was not the issue here. We countered this by having a 20% validation split while training so we could track how the model was doing on unseen data. These loss curves were also tracked for signs of overfitting: increase in validation loss even though training loss continues to drop. The loss curves were consistently convergent, so the model didn't really suffer from overfitting.

4.4 Practical Insights and Model Trustworthiness

4.4.1 Text Generation Process

The text generation process in our LSTM model is based on the sequential nature of the LSTM architecture. The input sequence is processed by the encoder that compresses the information into a context vector, passed to the decoder to generate the output sequence. It is a very deterministic procedure in terms of the probabilities assigned to each token in the output sequence, and thus the model can generate coherent responses based on the input data.

As seen in the results, the model was able to produce meaningful responses for a wide variety of inputs. This indicates that the model has successfully learned patterns in the data and, therefore, can respond appropriately in various contexts. One of the key factors behind the trustworthiness and utility of the model in practical applications like chatbots is its ability to generate coherent text.

4.4.2 Possible Improvements for Transparency

While our model performs at a high quality in generating responses, there are ways in which it can be further improved with regard to clarity. For instance, the attention mechanisms can be integrated into the system to visually highlight the most important parts of the input sequence on which the model is concentrating while generating the response. One area of improvement would be adding a feature to visualize how the model decides at every stage of text generation, making it better understand the basis for its conclusions.

Another step would be further analysis regarding how the model handles vague or out-of-context input, which are typical when it comes to real world interactions. Adding error analysis and improving the way this model handles those cases makes the model more reliable as well as transparent in all aspects of its operation.

5. Conclusion and Discussion

We hereby present this study, that endeavors to build a mental health chatbot using an encoder-decoder model trained on data provided through the book Happiness Unlimited: Awakening With Brahmakumaris by BK Shivani and Suresh Oberoi. The idea was to use this concept to create an effective chatting machine that responds to questions related to topics of interest in mental well-being for users, conveying pertinent and supporting responses.

Our approach leveraged sequence-to-sequence (seq2seq) models with LSTM-based architecture, which were trained on a carefully curated dataset of mental health-related conversations. The

model was able to generate relevant responses based on user inputs, with the chatbot continuously learning from the data to improve its responses.

We used performance metrics like accuracy and loss in addition to validation metrics while training the model to assess the effectiveness of the chatbot. The validation metrics enabled us to monitor the performance of the chatbot so that the model was generalizable and not overfitting or underfitting.

Through the training and testing process, we also derived insights into the behavior of the model, specifically how it handled text generation. This generation process was well linked with the probability distribution of tokens and the coherence sequence produced by the decoder. This kind of behavior let the model generate contextually relevant responses, though also showed a lot of places for improvement, especially conversational flow and diverse queries from the users.

Although promising results were found, several areas for future work were identified. First, the model would benefit from more training on a more diverse set of mental health-related conversations to improve its conversational depth. In addition, real-time user feedback in the training process can improve the chatbot's ability to provide personalized advice. The interpretability of the model could be improved by introducing more advanced explainability techniques for users and developers to better understand how the chatbot creates responses.

In conclusion, although the model has promise as a mental health chatbot, it still requires further development to make it more reliable, accurate, and able to respond to complex queries from users. Focusing on perfecting the flow of conversation, improving training data, and increasing transparency will continue to improve the ability of the chatbot to support users in a meaningful and compassionate way.