International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Emotion Detection Using a Bidirectional Long-Short Term Memory (BiLSTM) Neural Network

Amadi, Christian O.[a], Odii, Juliet N.[a], Okpalla, Chidimma. L[a], Ofoegbu Christopher I.[a]

[a] *Department of Computer Science, School of Information and Communication Technology, Federal University of Technology, Owerri, Nigeria*

ABSTRACT

Emotion detection in the context of chatbots holds immense promise for creating more empathetic and responsive conversational agents. This study presents a novel approach to enhancing chatbot capabilities by integrating emotion detection using Bidirectional Long Short-Term Memory (BiLSTM) neural networks. The primary objective of this research is to equip chatbots with the ability to discern and adapt to the emotional states of users during interactions. Leveraging the advantages of BiLSTM, we develop a model that can capture the temporal dependencies and contextual nuances in user messages, enabling it to accurately identify emotions such as happiness, sadness, anger, and more. The chatbot's architecture is augmented with the emotion detection module, allowing it to continuously analyze user input and provide emotionally tailored responses. Through a comprehensive dataset of conversational exchanges enriched with emotional labels, our model is trained to understand the intricacies of emotional expressions within text. The results of our experiments demonstrate the efficacy of the BiLSTM-based emotion detection approach within the chatbot framework. Users experience more personalized and empathetic interactions as the chatbot adapts its responses to match the detected emotional states. Comparative evaluations against traditional rule-based and non-emotion-aware chatbots underscore the significant improvements in user engagement and satisfaction. In conclusion, this research represents a significant advancement in the field of conversational Artificial Intelligence (AI). The integration of BiLSTM-based emotion detection empowers chatbots to better understand and respond to users' emotions, enhancing user experiences across a range of applications, from customer support to mental health companions. This work paves the way for more emotionally intelligent and empathetic AI-driven conversations, ultimately improving the quality and effectiveness of human-computer interactions.

Keywords: BiLSTM; chatbox; emotion; detection; neural network

1. Introduction

Emotion detection plays a crucial role in various applications, including sentiment analysis, customer feedback analysis, human-computer interaction, and mental health monitoring. Accurate identification of emotions from text or speech data can lead to improved user experiences and better decision-making in many domains. While traditional machine learning methods have shown some success in emotion detection, deep learning techniques have recently gained prominence due to their ability to capture complex patterns and relationships in data.

This paper focus on using Bidirectional Long Short-Term Memory (BiLSTM) neural networks for emotion detection. BiLSTMs are a type of recurrent neural network (RNN) architecture that can capture sequential information effectively by processing input data in both forward and backward directions. This bidirectional processing allows the model to consider the entire context of the input sequence, making it well-suited for tasks like emotion detection, where context is crucial. To enhance the performance of Bi-LSTM models in emotion detection, researchers have explored various techniques. Attention mechanisms, as those proposed by (Bahdannan et.al, 2014) have been integrated with Bi-LSTM models to enable them to focus on relevant parts of the input sequence, improving their ability to capture emotional cues. Additionally, transfer learning approaches have been employed to address the issue of data scarcity in emotion detection. In (Yang et. al, 2018) demonstrated the effectiveness of fine-tuning pre-trained Bi-LSTM models on large-scale datasets, leveraging knowledge from related tasks to improve emotion detection performance. Similarly, (Zhuang, 2014) investigated the use of Bi-LSTM models for emotion detection in social media data, considering the challenges posed by short and informal text.

To enhance the performance of Bi-LSTM models in emotion detection, researchers have explored various techniques. Attention mechanisms, such as those proposed by (Bahdannan et.al, 2014), have been integrated with Bi-LSTM models to enable them to focus on relevant parts of the input sequence, improving their ability to capture emotional cues. Additionally, transfer learning approaches have been employed to address the issue of data scarcity in emotion detection. Yang et. al, (2018) Demonstrated the effectiveness of fine-tuning pre-trained Bi-LSTM models on large-scale datasets, leveraging knowledge from related tasks to improve emotion detection performance. Despite significant advancements in emotion detection research, accurately recognizing and classifying emotions from textual data remains a challenging task. Traditional machine learning methods often struggle with capturing the complex sequential patterns and dependencies present in text data. Bi-directional Long Short-Term Memory (Bi-LSTM) networks have shown promise in various natural language processing tasks, but their effectiveness and potential limitations in emotion detection need to be explored further.

This research therefore, attempts to improve an existing emotion detection system in terms of addressing gaps and inefficiencies in its text pre-processing to produce better predictions. As yet, there has been no independent study conducted on the chosen existing system that is available online. It is therefore on this premise that this study becomes imperative to make contribution to the constant attempt to improve emotion detection systems driving towards achieving a system that accurately classify text inputs into thirteen emotion categories. The objective of this research is to develop a Bi-LSTM model to accurately classify text inputs into Thirteen emotion categories (Joy, anger, sad, relief, boredom, happiness, hate, fun, love, worry, enthusiasm, empty, neutral, and surprise.) using dataset from data world; to achieve an accuracy of at least 80% in classifying emotions using the F1-score as the evaluation metric; to integrated a BiLSTM-based emotion detection model into a chatbot, allowing the chatbot to identify and respond to user emotions in real-time.

The relevant of this research is that Emotions can be complex and influenced by a combination of factors, including linguistic cues, syntactic structures, and semantic relationships. These complexities in emotions can be improved if the emotion is classified in several folds by application of a suitable computer program driven by a portable Long Short-Term Memory algorithm. Accurate emotion prediction has significant practical implications in the field of Natural Language Processing and emotion detection such as enhancing customer experience, improving sentiment analysis, and enabling personalized services in organizations.

While there are existing a good number of implemented emotion detection systems and modifications with Natural Language Processing (NLP) communities, there is still need for constant improvements on these systems in order to sufficiently accommodate the growth of information available especially from sources on the internet. The natural language processing community and emotion detection community continue to research and seek to improve Bi-LSTM models, hoping to achieve a high level of scalability and adaptability. Achieving this will add to the series of milestones already attained in the research community.

Another significance of the paper is that it will lay a good foundation for customer experience and feedback analysis by identifying emotions expressed in customer reviews, comments, and social media posts. Organizations can give insights into customer satisfaction, identifying pain points, and address customer concerns. Emotion detection enables businesses to enhance customer experience, tailor their offerings and build stronger relationships with their customers. A highlight of points to justify the proposed study includes: the use of emotion detection lay a good foundation for customer experience and feedback analysis; The use of emotion detection gives insights into customer satisfaction; Emotion detection systems are used by organizations to build stronger with their customers, Comparing RNN, LSTM algorithms are effective in emotion detection.

2. Related Literature

Emotion Detection is famous recently because of the need in the health care sector. Emotion-sensing technology can facilitate communication between machines and humans. It can also help to improve the decision-making process. Many Machine Learning Models have been proposed to recognize emotions from the text. However, our focus is on the Bidirectional LSTM Model. Bidirectional LSTMs in short Bi-LSTM are an addition to regular LSTMs which are used to enhance the performance of the model on sequence classification problems. Bi-LSTMs use two LSTMs to train on sequential input. The first LSTM is used on the input sequence as it is. The second LSTM is used on a reversed representation of the input sequence. It helps in supplementing additional context and makes our model fast (Bahdannan et.al, 2014) Prior research on emotion detection has explored various techniques, including rule-based approaches, traditional machine learning algorithms, and deep learning models. Recent advancements in deep learning have led to improved performance in this task. Some of the notable deep learning architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models. However, the use of BiLSTMs specifically for emotion detection has received limited attention in the literature. In the framework of neural networks, BILSTM can quickly learn new feature representations from data within several hours. This is the main reason why we chose BI-LSTM to identify and classify emotions in this work (Zhuang, 2014)

2.1 Data Processing Method: Deep Learning

Deep learning is a research direction in the field of machine learning and the most popular research topic in the field of machine learning. The concept of deep learning comes from the study of how the human brain learns knowledge (Hasan et.al. 2014). By simulating the multi-layer neurons of the human brain to abstract concepts and interpret the mechanism of data, neural networks are established to realize the automatic feature extraction of input data (Ren et.al., 2016). The advantage of deep learning is that by layer-by-layer extraction, lower-level features can be abstracted into more expressive high-level features, and data features are extracted through hidden layer nodes. Therefore, deep learning is widely used in data analysis in various fields. It has been a great success in image classification, video analysis, computer vision, speech recognition, natural language processing, and many other fields (Felbo et.al., 2017). That is why our experiment chose to use deep learning to deal with emotion detection. What distinguishes deep learning evidently from traditional pattern recognition is the application of automatic learning based on big data. The special characteristics make it possible that the outcome with pattern recognition approach would be promoted (Zhang et.al. 2018). Over the past few decades, the characteristics of manual design have been dominant in a variety of applications for pattern recognition (Chen et. al, 2020). Because manual design relies primarily on the designer's prior knowledge and the characteristics of big data, big data does not take advantage of itself. The number of parameters allowed to appear in the design of features is minimal because of the reliance on manual tuning parameters. Almost all deep learning models can automatically learn the representation of features from big data and deep neural network can contain thousands of parameters. The model we chose this time is BI-LSTM. There is an apparent contrast between human learning and machine learning: manual design often takes five to ten years to design useful features by hand. However, in the framework of neural networks, BILSTM can quickly learn new feature representations from data within several hours. This is the main reason why we chose BI-LSTM to identify and classify emotions in this work (Zhuang, 2014).

2.2 Transfer Learning Model

Universal Sentence Encoder (USE) [10] is a deep neural network to create universal sentence embeddings. Universal embeddings are pretrained embeddings obtained from training deep learning models on a huge corpus. These pertained (generic) embeddings can be used in a wide variety of NLP tasks including text classification, semantic similarity and clustering. The Universal Sentence Encoder is trained and optimized for greater-than-word length text, such as sentences, phrases or short paragraphs. The input is a variable length text. The Universal Sentence Encoder encodes the input text into 512 dimensional embeddings. The USE embeddings are trained on different data sources and tasks with the aim of dynamically accommodating a wide variety of natural language understanding tasks which require modeling the meaning of word sequences rather than just individual words.

Essentially, there are two versions of the USE models. The first version makes use of a Deep Averaging Network (DAN) where input embeddings for words and bi-grams are first averaged together and then passed through a feed-forward deep neural network (DNN) to produce sentence embeddings (Cer et.al. 2018). Deep Averaging Network (DAN) is simpler than the second version. The primary advantage of the DAN encoder is that its compute time is linear in the length of the input sequence.

The second version makes use of the transformer-network based sentence encoding model. The transformer encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network, followed by layer normalization (Cer et.al. 2018). The results demonstrate that the transformer-based encoder achieves the best overall transfer task performance. However, this comes at the cost of computing time and memory usage scaling dramatically with sentence length. For our emotion classification task, we use the transformer-based encoder as it achieves better overall performance than the DAN encoder (Cer et.al. 2018). The LSTM algorithm leveraging prior knowledge from pre-trained embeddings to solve our emotion classification task. Our sentence embedding sub-network leverages the Universal Sentence Encoder. Their work fine-tunes the sentence embeddings using our collected emotion-labeled dataset.

Zhang et. al, (2018) build a feed-forward neural network with two hidden dense layers and the rectified linear activation function (ReLU). ReLU overcomes the vanishing gradient problem. This is good for deep neural networks which suffer from the vanishing and explosion gradient problem . A dense layer provides learning features from all the combinations of the features of the previous layer. The input of our model is 512-feature vectors created using the Universal Sentence Encoder technology. The resulting vector is then fed into fully connected layers culminating in a Softmax layer. We then fine-tune our model using collected labeled tweets introduced in Section V-A. We fine-tune the embedding weights by setting the trainable parameter to true. Here we leverage transfer learning in the form of pre-trained embeddings (Zhang et.al.2016)

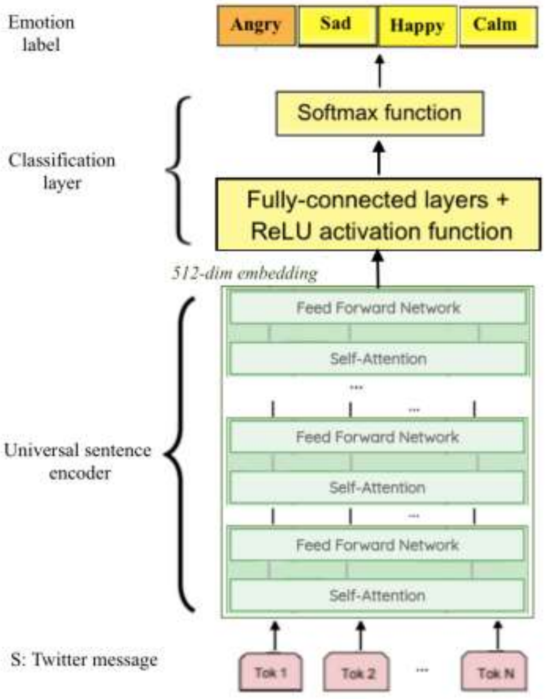


Fig. 1: Model of DeepEmotex-USE to classify emotion in text messages using Universal Sentence Encoder (Zhang. Et al.2018).

The overall architecture of DeepEmotex-USE is shown in Fig. 1. The input of the model is a twitter message. First, the embedding layer uses the pre-trained USE model to map a sentence into its embedding vector. The model that we are using splits the sentence into tokens, embeds each token and then combines them into context-aware 512-dimension embeddings. Then, the embeddings are passed through a feedforward neural network with ReLU activation. It projects the input into 256-dimension embeddings and feeds them to the classification layer to produce a classification probability. The output of our model is an emotion classification label. The main objective is to correctly predict the emotion of each tweet Zhang et,al. 2018)

2.3 BERT

A Transfer Learning Model using Bidirectional Encoder Representations from Transformers The model architecture of DeepEmotex-BERT using Bidirectional Encoder Representations from Transformers (BERT) is shown in Fig. 2. The input of the model is a twitter message and the output is an emotion label. We use the pretrained BERT model (Deylin et.al 2019) to generate text representations. BERT learns text representations using a bidirectional Transformer encoder pre-trained on the language modeling task. Transformers have a sequence-to-sequence model architecture. Each transformer includes a separate encoder and decoder component. The difference is in their use of attention known as self-attention. The core architecture consists of a stack of encoders fully connected to a stack of decoders. Each encoder consists of a self-attention component and a feed forward network. Each decoder consists of a self-attention component, an encoder-decoder attention component, and a feed forward component (Vaswari et.al. 2017)

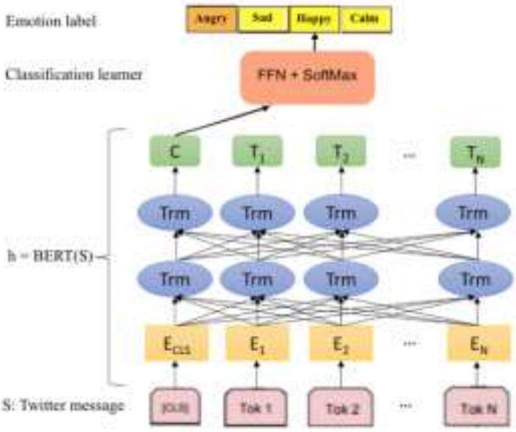


Fig. 2. Model of DeepEmotex to classify emotion in text messages using pretrained BERT (Vaswari et.al. 2017).

2.4 Bidirectional Recurrent Neural Networks Methods

Recurrent Neural Networks, or RNNs, are a specialized class of neural networks used to process sequential data. Sequential data can be considered a series of data points. For instance, video is sequential, as it is composed of a sequence of video frames; music is sequential, as it is a combination of a sequence of sound elements; and text is sequential, as it arises from a combination of letters. Modeling sequential data requires persisting the data learned from the previous instances. For example, if you are to predict the next argument during a debate, you must consider the previous argument put forth by the members involved in that debate. It enables user forms arguments are formed such that it is in line with the debate flow. Likewise, an RNN learns and remembers the data so as to formulate a decision, and this is dependent on the previous learning (Grave et.al., 2017)

Unlike a typical neural network, an RNN doesn't cap the input or output as a set of fixed-sized vectors. It also doesn't fix the number of computational steps required to train a model. It instead allows us to train the model with a sequence of vectors (sequential data).

Interestingly, an RNN maintains persistence of model parameters throughout the network. It implements Parameter Sharing so as to accommodate varying lengths of the sequential data. If we are to consider separate parameters for varying data chunks, neither would it be possible to generalize the data values across the series, nor would it be computationally feasible. Generalization is with respect to repetition of values in a series. A note in a song could be present elsewhere; this needs to be captured by an RNN so as to learn the dependency persisting in the data. Thus, rather than starting from scratch at every learning point, an RNN passes learned information to the following levels (Bahdanan et.al 2014)..

To enable parameter sharing and information persistence, an RNN makes use of loops.

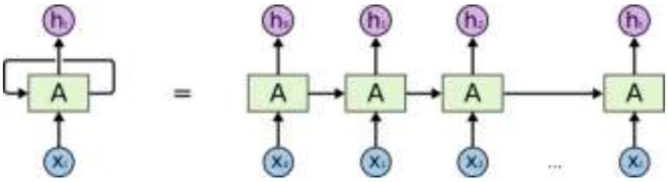


Fig. 3. Unfolding an RNN (Bahdanan et.al 2014).

Recurrent neural networks (RNN) have recently been trained successfully for time series modeling, and have been used to achieve state-of-the-art results in supervised tasks including handwriting recognition and speech recognition (Mohammed et.al. 2018). RNNs have also been used successfully in unsupervised learning of time series. Recently, RNNs have also been used to generate sequential data in a machine translation context, which further emphasizes the unsupervised setting. (Bahdanan et.al 2014 used a bidirectional RNN to encode a phrase into a vector, but settled for a unidirectional RNN to decode it into a translated phrase, perhaps because bidirectional RNNs have not been studied much as generative models. Even more recently, deep bidirectional RNN in speech recognition, generating text as output. Missing value reconstruction is interesting in at least three different senses.

Firstly, it can be used to cope with data that really has missing values. Secondly, reconstruction performance of artificially missing values can be used as a measure of performance in unsupervised learning. Thirdly, reconstruction of artificially missing values can be used as a training criterion (Goodfellow et.al., 2016).

While traditional RNN training criterions correspond to one-step prediction, training to reconstruct longer gaps can push the model towards concentrating on longer-term predictions. Note that the one-step

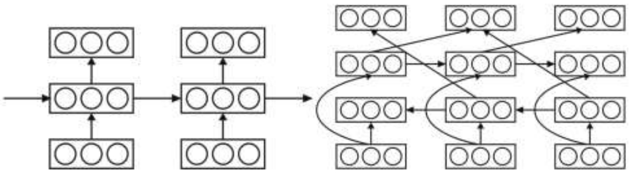


Fig. 4. Structure of the simple RNN (left) and the bidirectional RNN (right) (Bahdanan et.al 2014).

Prediction criterion is typically used even in approaches that otherwise concentrate on modelling long-term dependencies. When using unidirectional RNNs as generative models, it is straightforward to draw samples from the model in sequential order. However, inference is not trivial in smoothing tasks, where we want to evaluate probabilities for missing values in the middle of a time series. For discrete data, inference with gap sizes of one is feasible - however, inference with larger gap sizes becomes exponentially more expensive. Even sampling can be exponentially expensive with respect to the gap size. One strategy used for training models that are used for filling in gaps is to explicitly train the model with missing data. However, such a criterion has not to our knowledge yet been used and thoroughly evaluated compared with other inference strategies for RNNs. In this paper, we compare different methods of using RNNs to infer missing values for binary time series data. We evaluate the performance of two generative models that rely on bidirectional RNNs, and compare them to inference using a unidirectional RNN. The proposed methods are very favourable in terms of scalability (Sutskever et.al., 2010).

Not all scenarios involve learning from the immediately preceding data in a sequence. Consider a case where you are trying to predict a sentence from another sentence which was introduced a while back in a book or article. This requires remembering not just the immediately preceding data, but the earlier ones too. An RNN, owing to the parameter sharing mechanism, uses the same weights at every time step. Thus, during backpropagation, the gradient either explodes or vanishes; the network doesn't learn much from the data which is far away from the current position (Sutskever et.al., 2010)

To solve this problem, we use Long Short-Term Memory Networks, or LSTMs. An LSTM is capable of learning long-term dependencies.

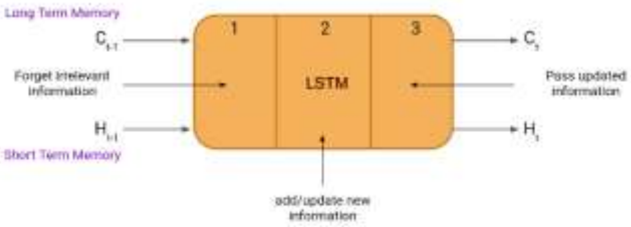


Fig. 5. Long Short-Term Memory Networks (Sutskever et.al., 2010)

Unlike in an RNN, where there's a simple layer in a network block, an LSTM block does some additional operations. Using input, output, and forget gates, it remembers the crucial information and forgets the unnecessary information that it learns throughout the network (Goodfellow et.al 2016).

2.5 Bidirectional LSTM

Bidirectional LSTM (BiLSTM) is a recurrent neural network used primarily on natural language processing. Unlike standard LSTM, the input flows in both directions, and it's capable of utilizing information from both sides. It's also a powerful tool for modeling the sequential dependencies between words and phrases in both directions of the sequence (Sutskever et.al., 2010). BiLSTM adds one more LSTM layer, which reverses the direction of information flow. Briefly, it means that the input sequence flows backward in the additional LSTM layer. Then we combine the outputs from both LSTM layers in several ways, such as average, sum, multiplication, or concatenation (Goodfellow et.al., 2016).

To illustrate, the unrolled BiLSTM is presented in the figure below:

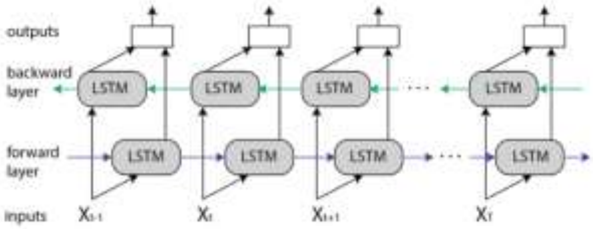


Fig. 6. Bi-LSTM illustration (Goodfellow et.al., 2016).

2.6 Challenges of Emotion Detection Using RNN

Mood detection using Recurrent Neural Networks (RNNs) poses several challenges that need to be addressed in order to improve the accuracy and robustness of the models. Here are some examples of challenges:

1. Limited Data Availability: One of the major challenges of mood detection using RNNs is the limited availability of labeled data. Collecting large amounts of labeled data for mood detection can be difficult and time-consuming, which limits the development and evaluation of RNN models. This challenge has been addressed by using data augmentation techniques, such as speech synthesis and data re-sampling, to increase the amount of labeled data available for training RNN models.

2. Inter-Subject Variability: Mood is a subjective experience that can vary significantly between individuals, which makes it difficult to develop RNN models that can generalize to different individuals. This challenge has been addressed by developing personalized RNN models that can adapt to the mood patterns of individual users (Yang et,al., 2019)

3. Variability in Input Modalities: Mood can be expressed through multiple modalities, such as speech, text, facial expressions, and physiological signals. However, these modalities can vary in their effectiveness for mood detection and may require different processing techniques. This challenge has been addressed by developing multi-modal RNN models that can integrate information from multiple modalities to improve mood detection performance..

4. Noisy and Incomplete Data: Mood detection using RNNs can be challenging due to noisy and incomplete input data, which can lead to incorrect or unreliable predictions. This challenge has been addressed by developing RNN models that can handle missing or corrupted data, such as using denoising autoencoders to reconstruct corrupted data (Zhang et.al., 2017)

Interpretability and Transparency: Mood detection using RNNs can also face challenges related to interpretability and transparency, as it can be difficult to understand how the model is making predictions. This challenge has been addressed by developing explainable RNN models that can provide insights into the model's decision-making process, such as using attention mechanisms to highlight important features in the input data (Yang et,al. 2019)

This research centered and focuses on emotion detection with the use of a Bi-directional Long Short-Term Memory Network. The study will address the challenge of accurately detecting and interpreting thirteen forms of emotions within the context of emotion detection using Bi-LSTM models. LSTM algorithm will be used to effectively handle and interpret this problem. Further, Bi-LSTM will be explored to improve the accuracy of emotion detection. Some limitation of this research is on hyperparameter search might not be feasible, leading to suboptimal performance.

3. Methodology

This paper developed a complete natural language processing pipeline, consisting of data wrangling and data pre-processing steps, followed by the creation of semantic embeddings and an LSTM based network. The emotional dataset used was imbalanced and has stop words. We created and utilized a basic pre-processing pipeline tasked with removal of special characters, punctuations, stop words and long repeating characters in a word. Punctuation removal was necessary as those characters do not offer much semantic meaning to the sentence. As the dataset comprises of tweets, some special characters present, such as - ‘@’,’#’ (resulting from mentions and hashtags) needed to be removed. There are also multiple occurrences of words with repeating characters - ‘hmmmmm’, ‘wowwwww’, as part of the tweets, so they had to be removed as well. This whole process was essential so as to avoid training the model on irrelevant data and adversely affecting its performance.

Again, an embeddings text was created as textual data that cannot be mathematically interpreted by machine learning algorithms, semantically correct embeddings for the text was also required. The paper opted for pre-trained embeddings for the model as the use case is very general and not specific to a certain type of corpus or task for example news corpus. In an emotion detection task, we would generally have text expressing general human feelings or emotions hence we decided to move ahead with pre trained models for vectorization of our corpus. The research came came across a number of available models but we decided to use pertained GloVe embeddings. GloVe is an unsupervised learning algorithm for producing vector representations for words. It is trained on global word to word statistics from a large text corpus, and therefore the resulting vectors show very interesting linear relations of the vector space.

The research used an existing data set on emotional data directly from dataworld and then used the downloaded Twitter emotion dataset to train a Bi-LSTM. The dataset consists of thirteen different classes: happiness, anger, sadness, relief, boredom, hate, fun, love, surprise, worry, enthusiasm, empty, neutral. We considered the tweets that are written in English. The dataset contains a total of 40,000 data samples. Table 1 Shows the distribution of data for different emotion categories.

Table1: Data Distribution among Different Classes

Classes	anger	relief	Bored	Happiness	hate	fun	Love	surprise	worry	Enthusiasm	sad	empty	Neutral
Size	8638	8459	5209	5165	3842	2187	1776	1526	1323	827	759	179	110

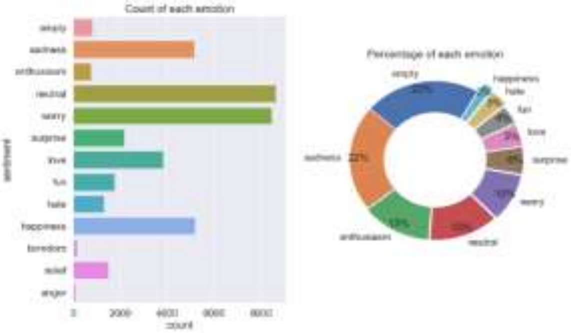


Fig. 7. Distribution of Sentiments and Check for Data

We notice that there's data imbalance, as some classes are very large (neutral, worry, happiness), while others are very small (anger, boredom, empty, etc). We proceed to applying data balancing technique. To balance the data, we have to use a method called up-sampling. upsample the other classes to match the largest class in dataframe (neutral). After balancing the data, we have as shown in Table 2.

Table 2: Data Distribution among Different Classes

Classes	anger	Relief	Bored	Happiness	hate	fun	Love	Surprise	Worry	Enthusiasm	sad	empty	Neutral
Size	8638	8638	8638	8638	8638	8638	8638	8638	8638	8638	8638	8638	8638

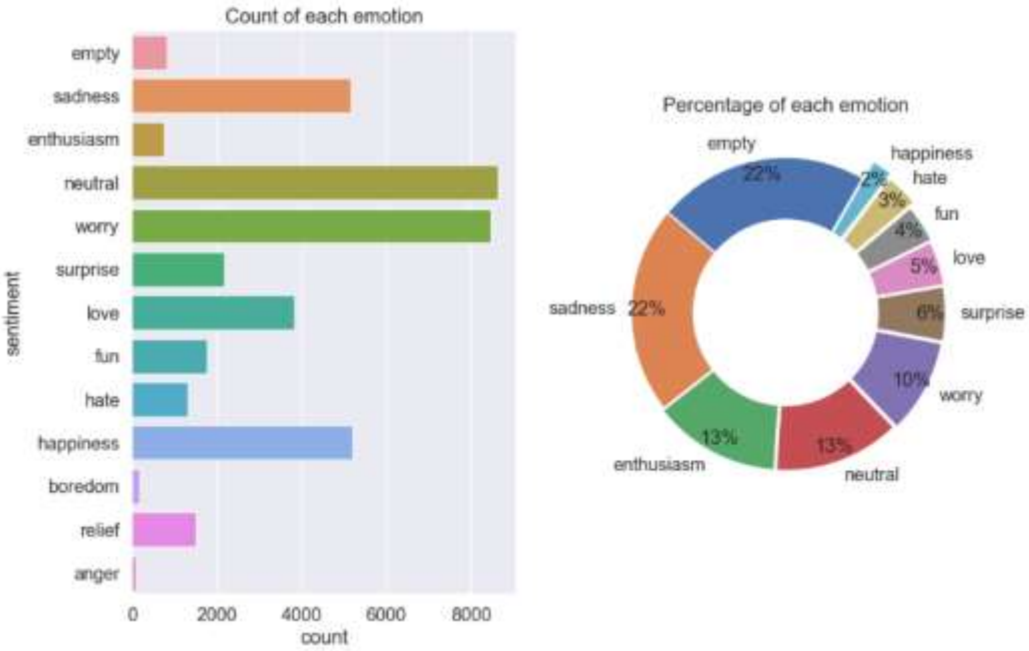


Fig. 8: Balanced Distribution after upsampling

3.1 Training and Hyper-Parameter Tuning

Traditional machine learning methods are trained using the sklearn library, while deep learning models are trained using the Keras library. Google Colab is used for all of the training. We used colab's GPU during training the LSTM models. Before training traditional machine learning models, we first find the embedding vector of each word of the text using the Global Vector (GloVe) for Word Representation or word embeddings and then stack the vectors of all the tokens/words to produce a matrix. The matrix is then flattened and used as the input for these models. The word embedding layer in deep learning models only takes the token id of the words/tokens as input and finds the word embedding. The weights are fine-tuned during training, and the embedding layer is initialized using the pre-trained word embedding.

3.2 The Proposed System

An improved system is proposed to achieve better emotion detection and empathy. The proposed model is pre-named 'MelBot' and is a stand-alone application which can display empathy and detect emotions in a more efficient manner. The core of emotion detection is in the pre-processing phase, which involves word and sentence tokenization as well as sentence scoring and ranking. Hence, the proposed model provides an efficient algorithm that can effectively handle these tasks and deliver a more reliable prediction. A major advantage of the developed model is in the manner in which it detects mood and empathize with the aim of producing meaningful responses. The research explore the possibility that an empathetic chatbot can buffer against the negative effects of social exclusion, particularly dampened mood.

Improving NLP Preprocessing Pipeline

To bridge the gap discovered in the preprocessing stage of the existing system, the study adopted a Bi-LSTM method. The Bi-LSTM layer consists of a cell state, an input gate, a forget gate, and an output gate. Bi-LSTMs are highly flexible and can be used for a wide range of NLP tasks, including sentiment analysis, Mood detection, machine translation, and text classification.

Implementing a Bi-LSTM Method in python

a. Prepare the data

b. Feature Scaling (Preprocessing of data)

c. Split the dataset for train and test

d. Converting features into NumPy array and reshaping the array into a shape accepted by the Bi-LSTM model

e. Build the architecture for the Bi-LSTM network

f. Use GloVe for the word embedding

g. Compile and fit the model (Training)

h. Evaluate the performance of the model

3.3 Proposed System Architecture

The proposed system architecture is presented in Fig. 9 and it describes in detail the structure of the emotion detection system. Based on the architecture, the source document is uploaded into the system via the input module. The input is processed and extracted and further undergoes some tokenization based on predefined LSTM algorithm. The final generated emotion recognition is delivered to the user.

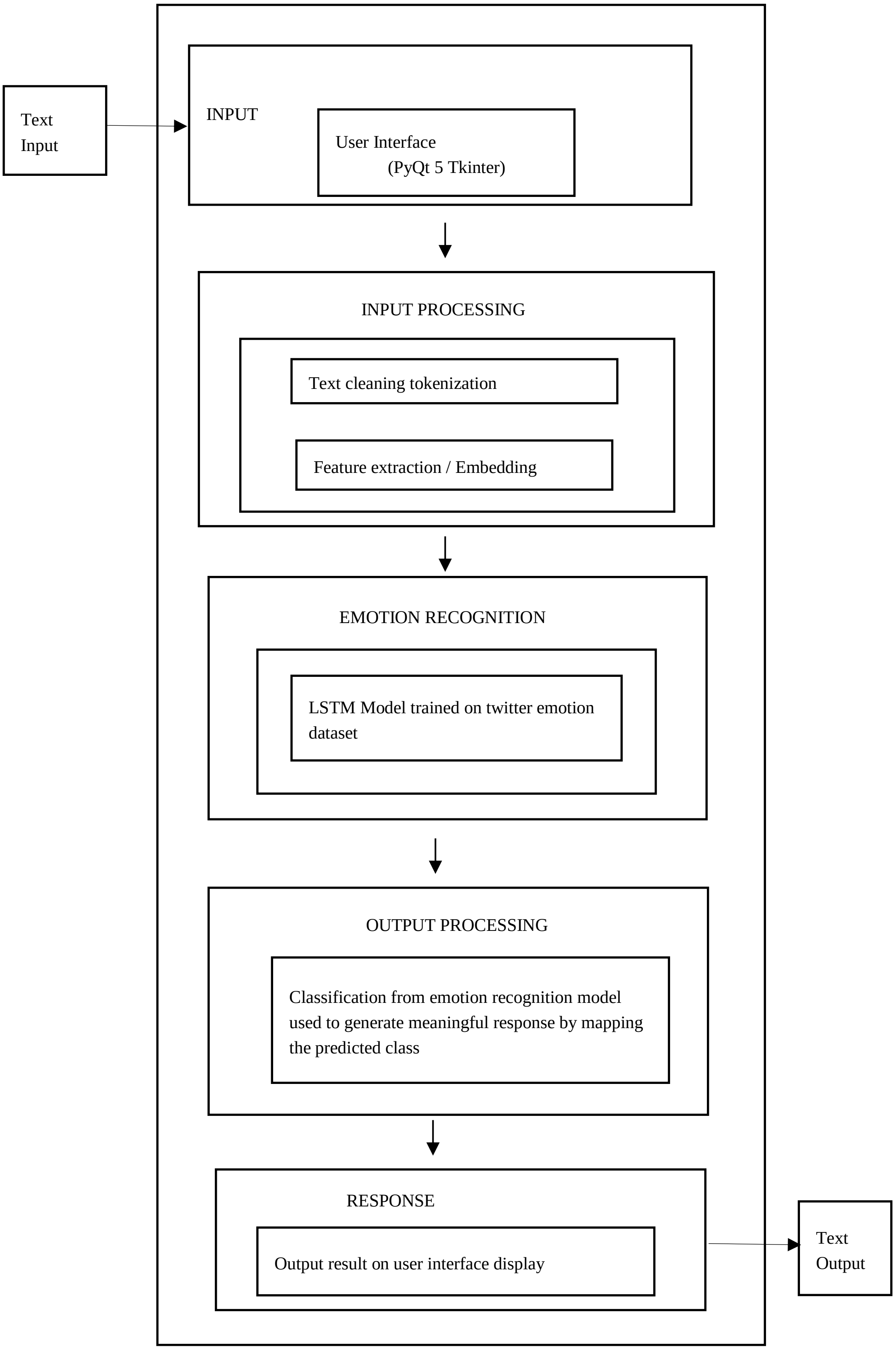

Text
Input
INPUT
User Interface
(PyQt 5 Tkinter)
INPUT PROCESSING
Text cleaning tokenization
Feature extraction / Embedding
EMOTION RECOGNITION
LSTM Model trained on twitter emotion dataset
OUTPUT PROCESSING
Classification from emotion recognition model used to generate meaningful response by mapping the predicted class
RESPONSE
Output result on user interface display
Text
Output

Fig. 9. Proposed System Architecture

The proposed software architecture is presented in Fig. 10 and it describes in detail the structure of the emotion detection system. Based on the architecture, the dataset is uploaded into the system to be pre-processed. The input is processed and extracted and further undergoes some embedding training based on predefined LSTM algorithm. The final generated emotion recognition is delivered to the user via the output module.

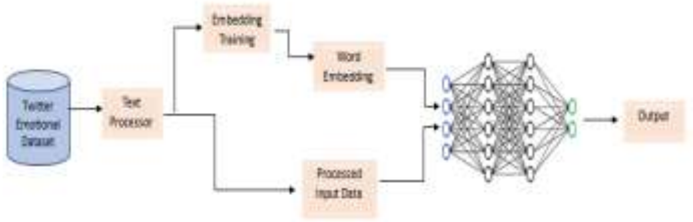
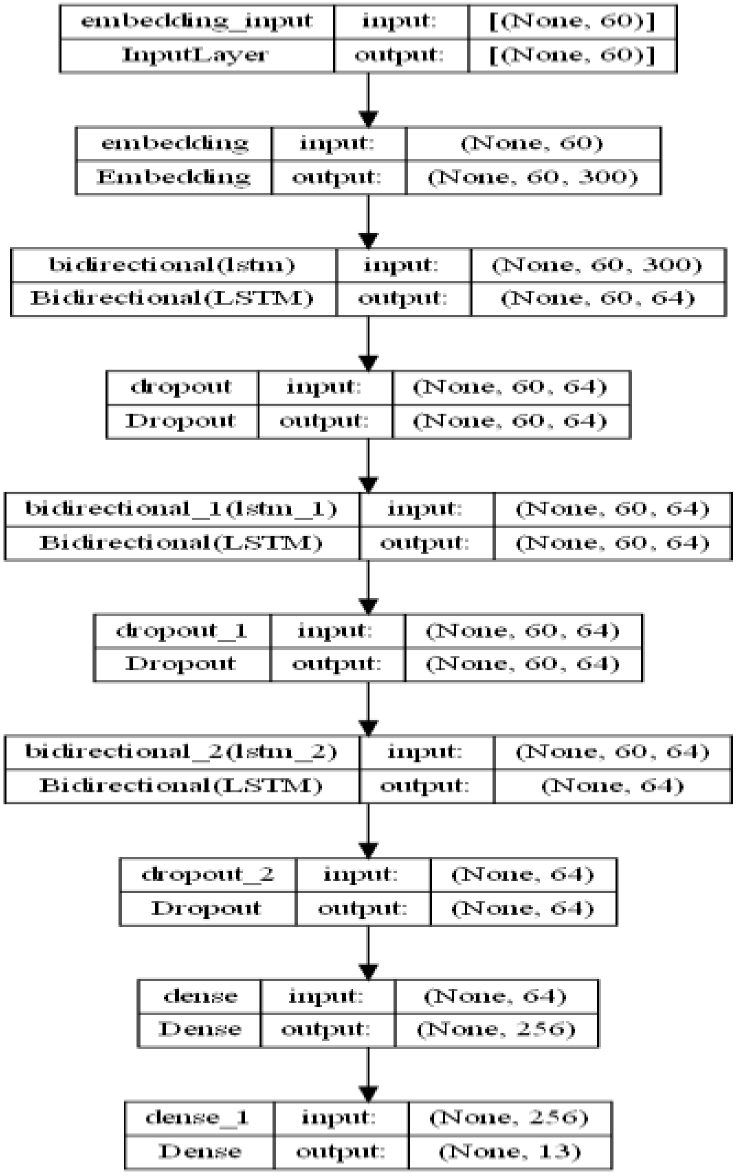


Fig.10. Proposed Software Architecture

3.4 Model Architecture for BiLSTM


embedding_input | input: | [(None, 60)]
InputLayer | output: | [(None, 60)]

embedding | input: | (None, 60)
Embedding | output: | (None, 60, 300)

bidirectional(lstm) | input: | (None, 60, 300)
Bidirectional(LSTM) | output: | (None, 60, 64)

dropout | input: | (None, 60, 64)
Dropout | output: | (None, 60, 64)

bidirectional_1(lstm_1) | input: | (None, 60, 64)
Bidirectional(LSTM) | output: | (None, 60, 64)

dropout_1 | input: | (None, 60, 64)
Dropout | output: | (None, 60, 64)

bidirectional_2(lstm_2) | input: | (None, 60, 64)
Bidirectional(LSTM) | output: | (None, 64)

dropout_2 | input: | (None, 64)
Dropout | output: | (None, 64)

dense | input: | (None, 64)
Dense | output: | (None, 256)

dense_1 | input: | (None, 256)
Dense | output: | (None, 13)

Fig. 11:Model Architecture for BiLSTM

3.6 Datasets and Evaluation Metrics

The research used one widely recognized emotion dataset - Twitter emotional Dataset. This dataset was chosen for its diversity and relevance in the field of emotion detection. The emotional dataset comprises of 40,000 sentences, out of which 40,818 are unique. There are thirteen emotion classes in the emotional dataset: joy, anger, sad, relief, boredom, happiness, hate, fun, love, worry, enthusiasm, empty, neutral, and surprise. The data set distribution in the dataset, as seen in Table 1. The table shows that the data set does not suffer from class imbalances. It indicates that there would be no need to implement strategies to mitigate class imbalances further in this data since it's already balanced. All the data samples in the dataset were used.

The data samples were cleaned to remove tags, double spacing, and some special characters observed to affect the detection's performance. The thirteen emotion classes were encoded to a numerical scale (i.e., 0, 1, 2, 3, 4, 5, 6,7,8,9,10,11,12) before the samples were split into two groups, i.e., 80% for training, 20% for testing. We set the maximum number of words to 75,000; this helped in setting a maximum sequence length of 60 for the tokenizer. It was done to ensure that no sentence is truncated as BiLSTM requires all supplied sentences to have the same size. The training and test samples were tokenized using the Natural Language Toolkit (NLTK) to generate the tokens. These tokens were encoded to create the input IDs, input mask, and segment id for each sentence denoted as input_ids, input_mask, and segment_ids, respectively. These were then fed to the tuned Bi-LSTM model as features for training and evaluation. The training set (i.e., trainFeatures) was provided to the adjusted Bi-LSTM model's input.

Experimental Setup: Model Training

The research trained a proposed BiLSTM-based emotion detection model on the emotional Dataset. The input text sequences were preprocessed as described in this paper. The model was trained for 5 epochs with a batch size of 128 using the rectified Adam as the optimizer. We employed sparse_categorical_cross entropy as the loss function during training.

.

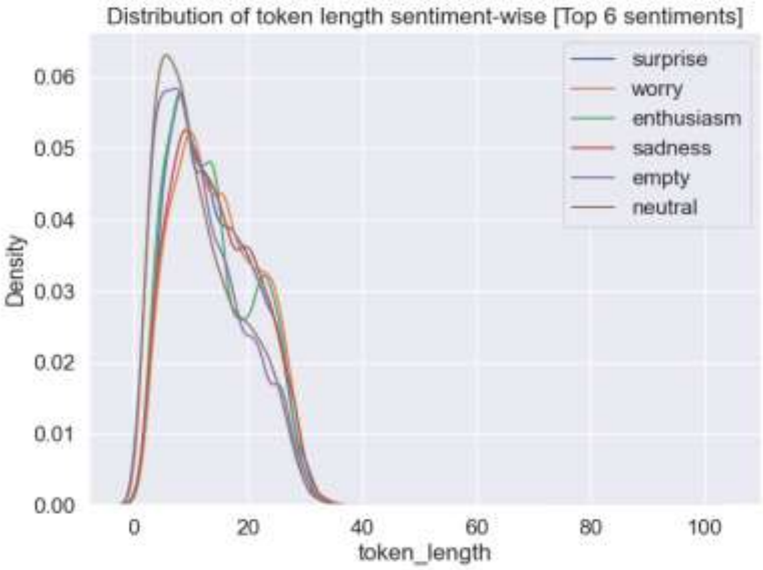


Fig. 12: Distribution of token length sentiment-wise

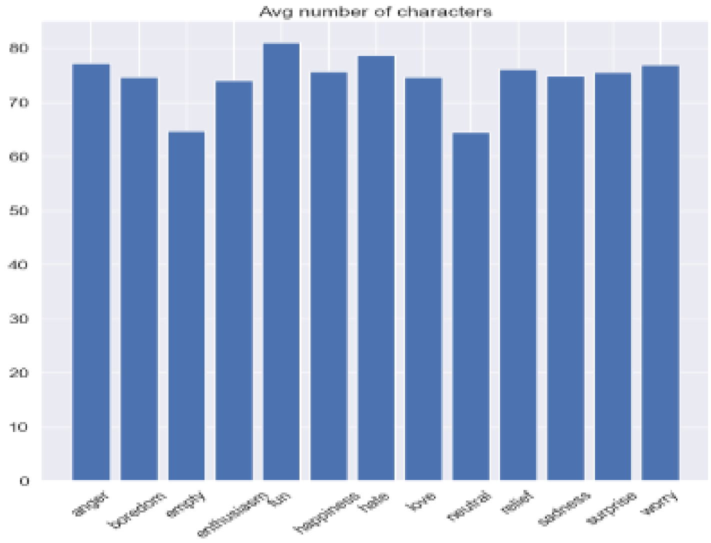


Fig.13. Distribution of token length sentiment-wise [Top 6 sentiments]

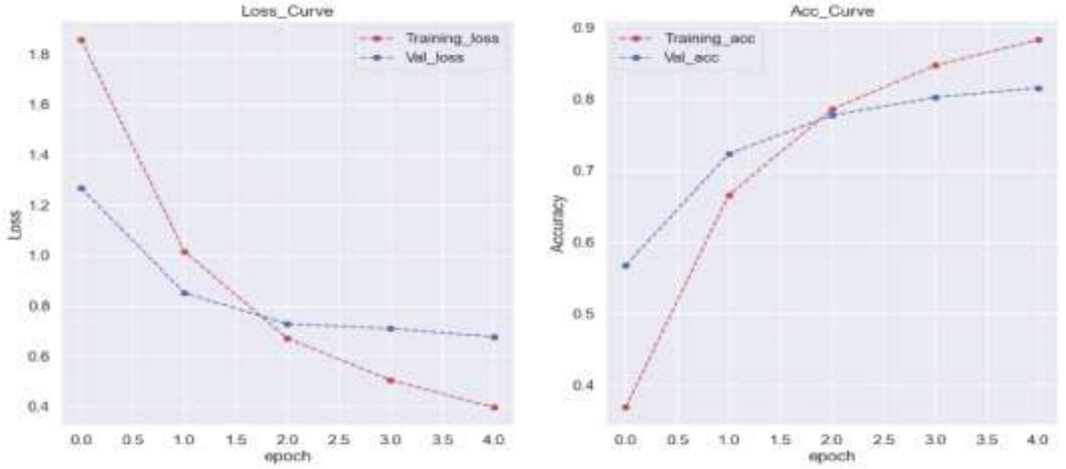


Fig. 15. Graph showing validation loss/accuracy and training loss/accuracy

Training and Testing Accuracy Plotting using RNN (BiLSTM)

Fig. 15. shows the testing and training accuracy plotting using BiLSTM model five(5) numbers taken from the epochs and batch size is 128 for calculating accuracy in both the models. Here epoch 702 value accuracy did not improve from 0.8823. Total test accuracy achieve by this model is 88.23%. With the increase of the "train epoch", the accuracy of the validation set gradually increases. As is shown in Fig. 16, with the increase of the "train epoch", the validation set loss gradually decreases; indicating that the BiLSTM model of the validation set gradually converges and has good stability. It could be shown that the BiLSTM model gradually converges in both the training and the validation set, with good stability.

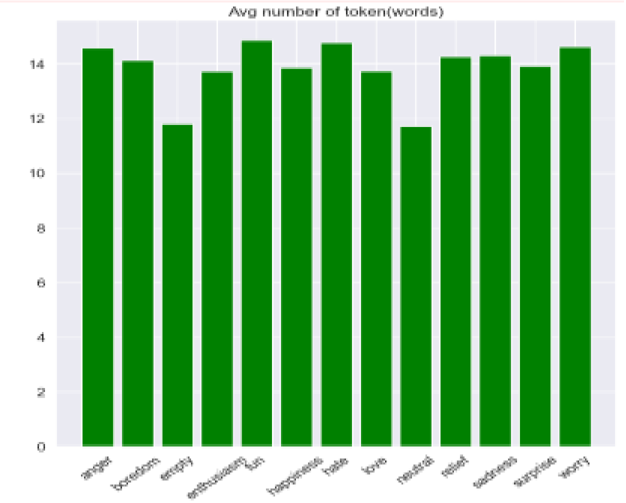


Fig. 16. Distribution of Average number of Token Length for each emotion class

Hyperparameter Tuning

To optimize our model's performance, we conducted hyperparameter tuning experiments, varying parameters such as the number of LSTM units, dropout rates, and learning rates. We selected the best-performing set of hyperparameters based on the validation results.

4. Results and Discussion

The results of the experiments are summarized in Table 3 shows the performance metrics on the trained Dataset. **4.3 Performance on Dataset**

From Table 3, it is observe that the BiLSTM-based model achieved an accuracy of 82% on the emotional Dataset. This demonstrates the model's ability to effectively detect emotions in an emotionally aware chatbot. The precision, recall, and F1-score are also competitive, indicating a good balance between correctly classifying positive and negative emotions. It further suggests that the model is adaptable and can generalize its emotion detection capabilities to different datasets. The precision, recall, and F1-score metrics further support the model's robustness.

Table 3: Evaluation of the Emotional dataset on the Bi-LSTM model

Classes	Precision	Recall	F1-Score	Support
Anger	1.00	1.00	1.00	1748
Relief	0.98	1.00	0.99	1711
Boredom	0.92	0.96	0.94	1739
Happiness	0.92	0.98	0.95	1682
Hate	0.94	0.88	0.91	1755
Fun	0.64	0.70	0.67	1732
Love	0.94	0.97	0.96	1722
Surprise	0.71	0.79	0.75	1738
Worry	0.45	0.39	0.42	1730
Enthusiasm	0.86	0.87	0.86	1727
Sad	0.69	0.69	0.69	1711
Empty	0.87	0.82	0.84	1767
Neutral	0.60	0.56	0.58	1697
Accuracy				
Macro Avg	**0.81**	**0.81**	**0.82**	**22459**
Weighted Avg	**0.81**	**0.82**	**0.81**	**22459**
			0.81	**22459**

4.1 Confusion Metrics

The confusion matrix provides a comprehensive view of the model's performance in detecting emotions

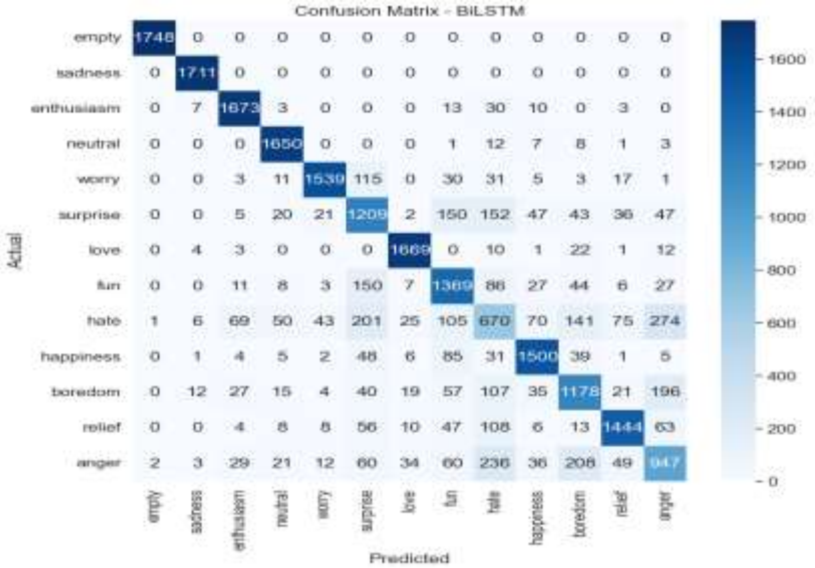


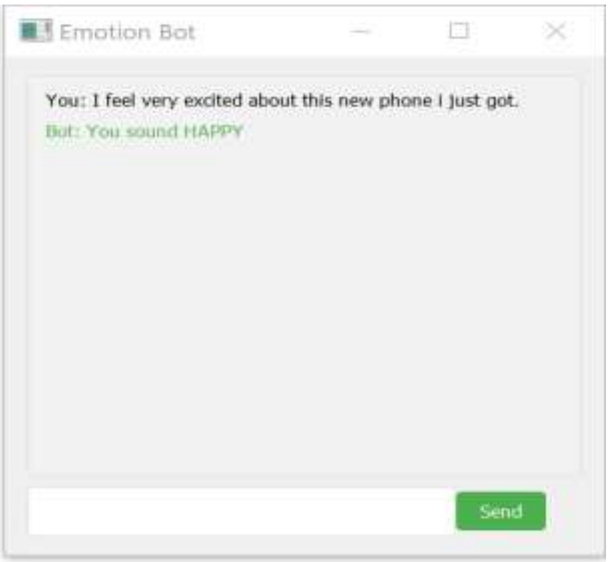
Fig.17. Confusion matrix table that summarizes the model's predictions versus the actual ground truth labels

Chatbot Performance Analysis

The chatbot's performance is promising, with high accuracy, precision, recall, and F1-Score. This indicates that the integration of emotion detection using BiLSTM has enhanced the chatbot's ability to recognize and respond to user emotions in real-time.

User Happy Mood Predicted in Real Time

After successfully integrating the BiLSTM model into a chatbot, Fig. 18 shows a chatbot that interacts with a user and accurately detected user’s emotion as ‘happy’ using a BiLSTM model. Some of its features of the interface includes; an input box and a send button.



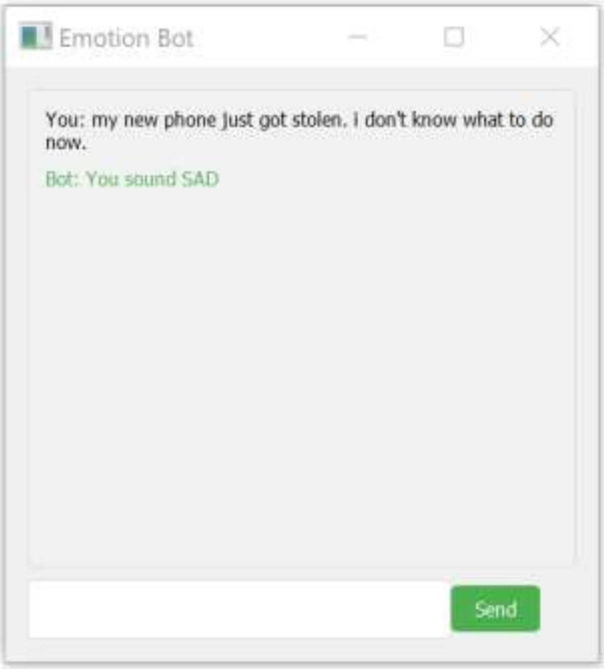
You: I feel very excited about this new phone I just got.

Bot: You sound HAPPY

Fig. 18. User Happy Mood Predicted in Real Time

User Sad Mood Predicted in Real Time

After successfully integrating the BiLSTM model into a chatbot, Fig. 19. shows a chatbot that interacts with a user and accurately detected user's emotion as 'Sad' using a BiLSTM model. Some of its features of the interface includes; an input box and a send button.



You: my new phone just got stolen. i don't know what to do now.

Bot: You sound SAD

Fig. 19. User Sad Mood Predicted in Real Time

User Angry Mood Predicted in Real Time

After successfully integrating the BiLSTM model into a chatbot, Fig. 19 shows a chatbot that interacts with a user and accurately detected user's emotion as 'Angry' using a BiLSTM model. Some of its features of the interface includes; an input box and a send button.

Discussion

The results of our research demonstrate the effectiveness of the BiLSTM-based emotion detection model in accurately identifying emotions in text data. The model's strong performance on both Dataset and chatbot suggests its potential for real-world applications in customer service, mental health support, and virtual companionship. The research further provides substantial evidence of the effectiveness of integrating emotion detection using BiLSTM in a chatbot. This technology has the potential to significantly improve user interactions with conversational agents by making them more empathetic and responsive to user emotions. However, it is important to recognize that the technology is not without challenges. Ensuring user privacy and data security, addressing ethical concerns, and dealing with situations where emotions are complex and nuanced are areas that warrant further consideration.

This paper has presented the experimental result and analysis of our research on integrating emotion detection using BiLSTM in a chatbot. The performance metrics confirm the effectiveness of our approach in enhancing the chatbot's ability to detect and respond to user emotions.

Conclusion

This research has presented a comprehensive exploration of emotion detection using BiLSTM in a chatbot, offering a new dimension to AI-driven conversational systems. Our research demonstrates the potential for emotion-aware chatbots to enhance user experience and engagement. As AI technologies continue to advance, the incorporation of emotional intelligence in conversational agents holds great promise for a more empathetic and user-centric digital future

References

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Yang M., Chen Ning. Covering song recognition model based on deep learning and hand-designed feature fusion[J]. Journal of East China University of Science and Technology (Natural Science), 2018, 44(05):138-145.

Wang, W., Huang, K., Li, C., & Li, X. (2020). Emotion classification of social media texts using bidirectional LSTM. IEEE Access, 8, 40299-40307.

Zhuang.Y (2014) Interpretation of Research Progress and Trends of Big Data from the Perspective of Database[J]. Electronic Technology and Software Engineering, 2014(17):206-206.

Hasan M, Rundensteiner, E. and Agu, E.(2014) "Emotex: Detecting emotions in twitter messages," in Proceedings of the Sixth ASE International Conference on Social Computing (SocialCom 2014). Academy of Science and Engineering (ASE), USA,

Ren, Y, Zhang, Y. Zhang, M. and Ji D.(2016), "Context-sensitive twitter sentiment classification using neural network." in AAAI, 2016, pp. 215– 221.

Felbo, B. Mislove, A., Søgaard, A., Rahwan, I. and Lehmann, I. S (2017) "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1615–1625.

Zhang, L., Wang, S.and Liu, B. (2018) "Deep learning for sentiment analysis: A survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1253

Chen, S Y. Hou, Y. Cui, W. Che, T. Liu, and X. Yu, "Recall and learn: Fine-tuning deep pretrained language models with less forgetting," arXiv preprint arXiv:2004.12651, 2020.

Cer, D. Y. Yang, Y., Kong, S., Hua, N. Limtiaco,N., John, R.S. Constant, N., Guajardo-Cespedes, M., and Yuan, S.. (2018) "Universal sentence encoder," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,

Devlin,J. M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 2018

Grave, A., Bojanowski, P. Joulin, E and Mikolov, T. (2017)"Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

Goodfellow, I., Bengio, Y., Courville (2016), A.．Deep learning (Vol. 1)：Cambridge：MIT Press：367-415

Vaswani, A.. Shazeer, N. Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser L., and. Polosukhin, I (2017) "Attention is all you need," in Advances in neural information processing systems, , pp. 5998–6008

Mohammad, S. Bravo-Marquez, F., Salameh, M and Kiritchenko, S . (2018) "Semeval-2018 task 1: Affect in tweets," in Proceedings of the 12th international workshop on semantic evaluation, pp. 1–17.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2019). Learning to attend, copy, and generate for session-based query suggestion. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 165-174).

Zhang, Y., Yuan, H., Wang, J. and Zhang, X. (2017) "Ynu-hpcc at emoint2017: using a cnn-lstm model for sentiment intensity prediction," in Proceedings of the 8th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp. 200–204.

Sutskever, I. and Hinton, G. (2010). Temporal-kernel recurrent neural networks. Neural Networks, 23(2):239– 243,