A Deep Learning Approach to Integrate Human-Level Understanding in a Chatbot

by

Amirul Islam Al Mamun
20241035
Afia Fairoose Abedin
17101360
Rownak Jahan Nowrin
17301002

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

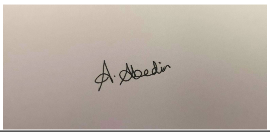
Declaration

It is hereby declared that

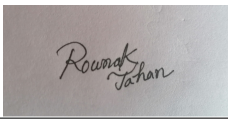
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Amirul Islam Al Mamun
20241035



Afia Fairoose Abedin
17101360



Rownak Jahan Nowrin
17301002

Approval

The thesis/project titled "Analysis to Develop a Human-Like Chatbot using Deep Learning" submitted by -

1. Amirul Islam Al Mamun (20241035)
2. Afia Fairoose Abedin (17101360)
3. Rownak Jahan Nowrin (17301002)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 11, 2021.

Examining Committee:

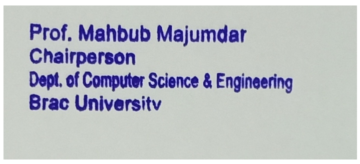
Supervisor:
(Member)

Dr. Amitabha Chakrabarty
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Dr. Mahbubul Alam Majumdar
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

AI-powered computers like chatbots have taken over the market today to reduce human workload. Unlike humans, chatbots reply immediately, are available 24/7 and can assist several people at the same time. Due to the outbreak of Covid-19[3] as everything has just shifted to online, the demand of bots has increased tremendously. Considering the various applications, it is estimated that the chatbot market will reach $1.25 billion by 2025[2]. Though chatbots perform well in task-oriented activities, in most cases they fail to understand personalised opinions, statements or even queries. Generally, people prefer human agents as it is easier to share personal views and feedbacks with them. Also, poor understanding capabilities of a machine disinterest humans to continue conversations with them. Usually, chatbots give absurd responses when they are unable to interpret a user's text accurately. Hence, it is very essential to develop chatbots with human-level understanding. Most bots are incorporated with sentiments to analyse reviews of products and services of an organisation. However, this is not enough as only positive and negative judgements cannot help an organization improve their lackings. To make chatbots function more precisely, it needs to identify the granular reaction of a customer as well as the reason behind it. Thus, in our research we incorporated all these key features that are necessary for a chatbot to have a human-like understanding of a text. We performed sentiment analysis, emotion detection, intent classification and name-entity recognition using deep learning to modify chatbots with humanistic understanding and intelligence. Conventionally, machine learning is used to perform analysis of the components mentioned above, however, it is seen that ML models ,often ,are unable to understand the inferences and complex sentences of human utterance and this is where deep learning has the upperhand [1]. Therefore, we chose deep learning models such as LSTM, Bi-directional LSTM, GRU, Bi-directional GRU etc to train our chatbot so that it can make more accurate predictions. From our training, we got the best performance model LSTM with accuracy 89% in sentiment analysis and Bi-directional GRU with accuracy 91%, 80.7%, 98.9% for emotion detection, intent classification and named-entity recognition respectively.

Keywords: Deep learning; Sentiment analysis; Emotion detection; Intent classification; Named-entity recognition; Humanistic Chatbot

Acknowledgement

Despite all the challenges and the extraordinary situation of COVID-19, we were able to successfully complete our thesis by the grace of Almighty. We are extremely thankful to our supervisor, Dr.Amitabha Chakrabarty who guided us and constantly monitored our work and motivated us to bring out the best possible outcome. His patience towards our constantly expanding and changing of ideas is really recommendable. Without his feedback and support our thesis wouldn't have been successful. We are also grateful to our co-supervisor, Moin Mostakim who has enlightened our knowledge in the field of deep learning and helped us understand the various applications of it. Furthermore, we are really honored to be guided by Sudip Kumar Nasker, Associate Professor in Jadavpur University, Kolkata, who briefed us regarding NLP and the possible approaches to work efficiently. We are also thankful to our faculty members who assisted us whenever needed and our friends who constantly encouraged our work. Lastly, we would like to appreciate each of us for dedicatedly working hard and keeping no stone untouched to make our research as qualitiful as possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In recent times, a large number of people are involved in establishing their own business. But, it is tremendously tough to stay updated with technology and hold a place in the market full of competition. Chatbot is one of the most advanced features incorporated in most organizations or online platforms today . As customer care teams have a lot of constraints to working hours, multiple responses at a time and efficiency, conversational agents definitely solve all the problems. Like humans, it doesn't get tired or make delays to reply. Chatbots handle similar questions by the customers, help firms in advertising their brands , are very cost-effective and also help organizations overcome their drawbacks by analysing customers feedback[4]. AI bots had predefined questionnaires before and the customers answered them which were later evaluated by the service team. Nowadays, chatbots have evolved by deep learning and they manage all conversations without the need of human support. Within a year, there has been 100k to 300k chatbots on messenger. Most popular brands are developing their very own chatbots[5]. Not only in business, but also applications of chatbots are emerging in medical fields as well. Chatbot applications are developed to imitate psychiatrist techniques to uplift a person's mood or reduce stress . Mental health bots like Woebot, Replica, Pacifica etc. all follow psychological approaches such as CBT (Cognitive Behavioural Therapy) to generate responses in a way that is more humanistic[6]. Hence, improving chatbot models can play a great role in blooming the era of virtual assistants and open the doors of vast effective applications.

1.2 Problem Statement

As previously said, conversational agents have been introduced to reduce the human effort. However, completely replacing a human is quite challenging. There are many conventional AI-bots in the market which generate fixed responses and thus can be monotonous at times for the user as it gives redundant replies. That's why even today most users prefer human agents to solve their problems. For a virtual agent to respond like a human, it needs to "understand" the problem of the user and provide "sensible" replies to humans[7]. But, it is extremely difficult to mimic a human behaviour, psychology and mind to create human-like conversation. It is also important, because no one wants to continue a conversation with a dead-end robot[6]. Even if we talk about one of the first realistic-bot, SIRI which is developed by Apple has a vast range of features helping users to manage calendar, make calls, schedule meetings and what not. However, if you have ever tried to talk more interactively, it fails to understand and most frequently replies "I am sorry I couldn' t understand. Could you try again?". The most significant drawback of a chatbot is, it lacks to understand the specific context of the sentences as well as the right sentiment that needs to be conveyed in response. Usually, it identifies the most similar keywords and responds to the closely matched text from its database. Eventually, most of the replies become inconsistent and meaningless. Furthermore, using a bot in the psychological field needs to be even more human-like giving the vibe of human-connection and empathy[6]. All these are only possible if the virtual assistant is able to get the right sentiment, emotion and intent of the person and clearly have an overall understanding of the user's situation. Once this problem can be solved, it can be used in every sector for personalised conversations and this information can be very helpful to the organizations whether to help a mental health patient or to improve various services and products sold. Even universities can use it to know the exact problem that a student is going through in a particular course that he failed. Therefore, through various research, we are trying to develop a model that will analyse a text and understand it just like a human so that making responses can be ever easier and accurate.

1.3 Aim to study

Analysing the various features that are required by a chatbot to gain a human-like understanding of text is the aim of our research. We are trying to develop a hybrid model that will understand the sentiment, emotion, named-entity and intent of the user's text and also will be able to identify the correct entities mentioned. We chose particularly deep learning because DL models help to extract the meaning of mixed contrary complex sentences given by a user. The layers in DL models creates a network that helps bots to learn accurately on their own which normal ML bots fail to do[8]. It is easy to generate a dialog without having a sense of it, even we humans do the same. At times humans speak out without knowing the reason behind the former person's opinion or statement. These kinds of conversation in real life itself are not healthy and should not be encouraged. Hence, to replicate an idealistic human nature, we need to build the chatbot with knowledge, consistency and empathy [9] so that users can engage in interactive conversations and find interest to share their views, emotions or motives. Our goal is to implement understanding capabilities in a chatbot which can be used for developing a human-like bot.

1.4 Research Methodology

We started our research by analysing a commonly used feature, sentiment, which is incorporated in most customer service chatbots to help the company rate their products. It is also essential in various movie reviews or story book feedback. We implemented various deep learning models from simple neural networks to LSTM and GRU. Based on accuracy, we selected the best model for our sentimental analysis. Next, through our study, we came up to classify sentiment into more meaningful expression that is emotion detection. We used 7 emotions of joy, sadness, disgust, shame, guilty, anger and fear to identify the specific mental state of the user. We applied Bi-LSTM and Bi-GRU for predicting the correct emotion. Next, we decided to bring out the intent of the conversation. To understand what the user wants or what he/she needs, we need to know the intent of the text. Similar to emotion detection, we used Bi-directional LSTM and Bi-directional GRU to experiment the intent classification. For getting more detailed intent, we performed named-entity recognition as well. Named-entity recognition is important because it identifies the person, place, organization, region or date the user is talking about. For example- *"What is the time in Bangladesh now?"* Here *"knowing time"* is the intent and "Bangladesh" is the entity. We used Bi-directional LSTM and Bi-directional GRU to recognize Named-Entity. As we have performed different analysis on different human understanding features it was quite impossible to find one single dataset to incorporate all the analysis. Thus, all our feature analysis was executed by independent datasets. As our domain was deep learning, we performed all our analysis within this field. Despite the unavailability of one dataset, we tried to integrate the analysed models so that all the features can be extracted from a single text.To give accurate results, Deep Learning bots need continuous training. So, it was quite obvious to get incorrect results initially. However, through our accuracy of models, estimations and various research we proposed, a text analysis structure that will give a chatbot multimodal understanding. The following Figure 1.1 shows the workflow of our executed analysis.

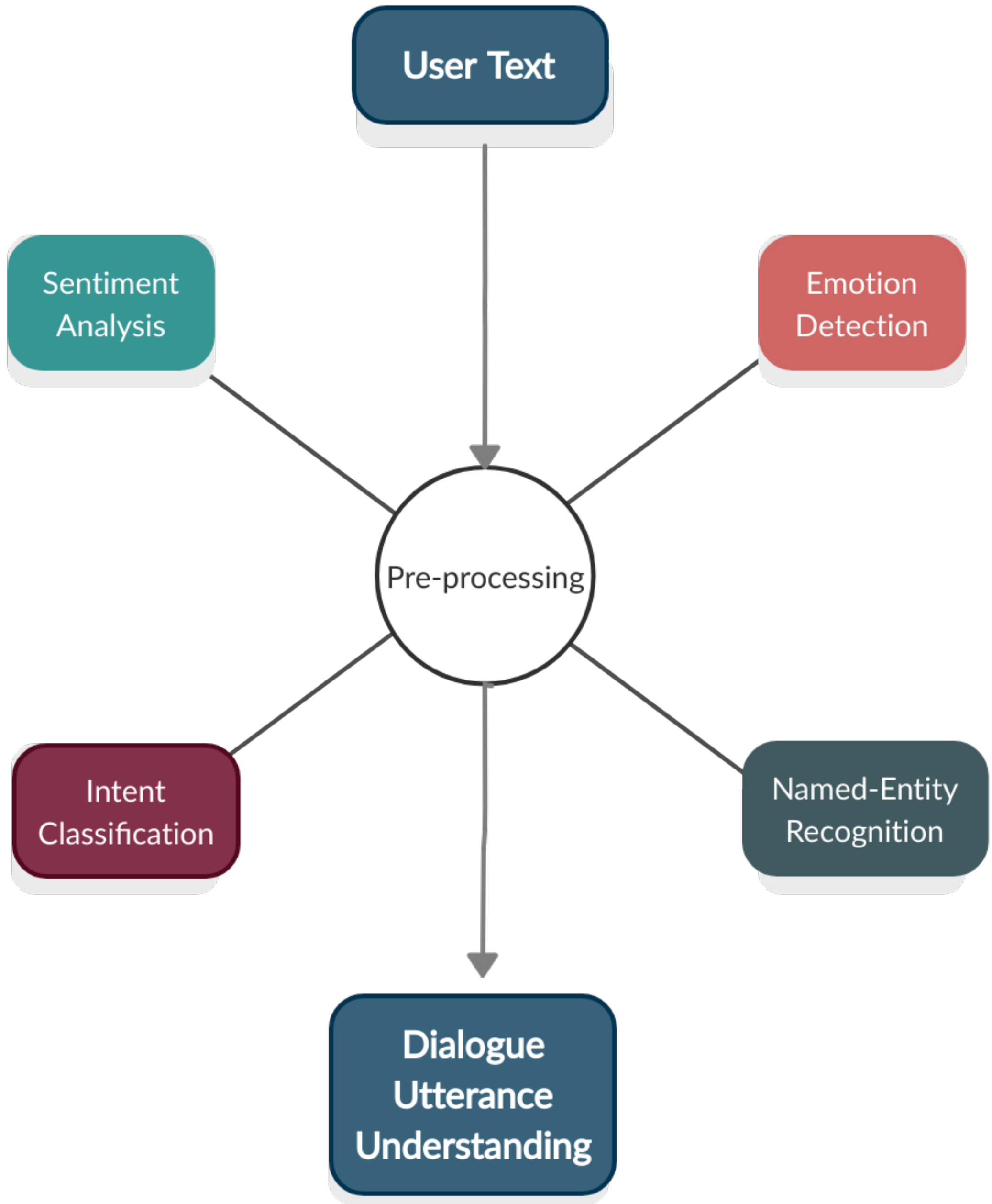


Figure 1.1: Analysis work flow for dialogue utterence understanding

1.5 Thesis Outline

Our thesis is concerned with identifying the best possible pathway to develop a human-level text interpretation by a virtual bot. The overall paper is a guideline to develop a more human-like chatbot.

The first part (Chapter1) of our paper is the introduction that states the various applications of chatbots, the drawbacks of it and finally about our work that focuses on making improvements to a better designed conversational agent.

Secondly, in our related works (Chapter 2) we mentioned about the research works similar to ours and how we are incorporating ideas and pitfalls from all of them to create our own model. We also added the advantages of deep learning that were studied by different researchers and how it helped them in achieving their objective.

Third is our model selection (Chapter 3) that will give a brief summary of different neural network models and reasons behind using it . As we used similar models in our different analysis, discussing the same mechanisms would be repetitive. Hence, this section will give you the background formulas and functions of each of the deep learning models.

Most important part of our paper is experimental analysis and results (Chapter 4) that elaborately explains our methodology and step by step analysis to reach our goal. It is the detailed explanation of how we gathered the data, pre-processed it and shaped it to fit in our models. It has images and graphs of our results as well as comparison that shows which model gives better accuracy. Moreover, after our analysis we showed visualizations of the predictions of all the features integrated together.

Finally, we concluded with our visions - a fully functional human-like conversational agent (Chapter 5).

Chapter 2

Related Work

To ensure the benefits of developing chatbots, we investigated the paper [10] that presented the different real-life applications of chatbots such as in education, automated telephone answering, e-commerce, help desk etc. The paper discusses ALICE chatbot which worked by training with chat-patterns in AIML. In addition, the Java program was used to translate machine readable text to AIML format. Chatbots are used for entertainment purposes. For example, ELIZA, which was used to mimic people by keyword matching. The bots replied on the basis of the topic of the keyword which was as input from the user. The author also added that it can be used for learning purposes. They referred to paper [12] that demonstrates an experiment for students to chat with the bot to improve their English language. It failed as the chatbot ALICE worked on keyword matching rather than understanding the statement or question made by the user. Later, the author suggested another paper [11] an intelligent Web-Based teaching system for foreign language learning was developed and it was successful. The results showed chatbot was able to help the students more efficiently than a study companion or teacher. It repeated the same contents required by the student without getting tired, answered questions from several topics and helped them to practice reading and listening too. Furthermore, the author concluded chatbots can be used as an information collecting tool too[10].

To know about previous works on sentimental analysis, we read the research paper [13] where text mining techniques were applied to find sentiment and emotion from twitter dataset. Twitter is one of the most famous social platforms where people express their views and emotions. Any big occurrence or event gets the lamplight on twitter. Hence, twitter has a huge amount of data that can be very helpful for knowing the overall impact on people's emotional state through a particular event. A sentiment file was created by incorporating frequently used words and their sentiment. Words from twitter were compared with the sentiment file using Bayes algorithm and given the corresponding sentiment. However, we can see the limitations of this work as it will not be able to predict sentiment of any word apart from the words stored in the database and therefore, we get a scope here to improve the drawbacks of their work[13].

Also, an interesting Bangla intelligent social robot was introduced in paper [14] which communicates in bangla analysing sentiments with the help of machine learning algorithms. Information on variables of interest are gathered through data collection. The authors have refined a huge amount of data to fit in their machine learning algorithm. Fictional conversation extracted from raw movie script were used as the metadata collection of the corpus. Data refining is done through cosine similarity. The system needed to refine 1000 movie dataset by selecting dialog serial and used the Google API to translate it. The authors also used Naive Bayes classification algorithm to train the dataset. The processing part of the system uses one logic adapter to select the closest matched dialogue for generation[14].

To work on context understanding of a chatbot, we went through the paper [15]which was a chatbot application based on NLP along with emotion recognition. The emotion recognition is done by determining the behavioural user pattern and contextual user pattern from 10 features which indicates the emotional state of the user. The system also aligned temporally the audio and visual streams through using the soft attention mechanism. The author also added emotion embedding vectors in the output layer of RNN. The system uses 4 levels of hierarchy to understand natural language input sentences and recognize the user's emotion. The response generation is done through collecting, analyzing and integrating input dialogues consisting of text. Then document intention extraction is done from the text by neural network. Database is used to store the response document.The system understands the user state and observation of continuous user's emotional changes sensitively through NLP [15].

Also, paper [16] proposed a chatbot for old people to decrease their loneliness and depression . The model is fed with data from MHMC chitchats which includes 40 different topics. The texts in the chit-chat database are sorted and generated into a pattern. The paper proposes two-layered LSTM to synchronise and find relation between the words and sentences. The final response is chosen based on comparing messages through Euclidean distance. The authors used open domain and hence, the bot is able to answer questions from various topics [16].

We gathered knowledge about LSTM by studying the advantages of using LSTM to overcome the vanishing gradient and long-term dependencies of RNN from paper [17]. The author has used a truncated gradient based algorithm so that there is a constant error flow. To make this happen, Naive approach is used where the constant error carousel, w_{jj} is set to 1 in the usual error calculation function. They have also mentioned that this approach arises two problems of input and output weight conflicts. The LSTM has a control gate which helps to overcome the above mentioned problem. This model has an input layer, hidden layer and output layer. The hidden layer generates the final state results using sigmoid functions. Furthermore, it also mentions the advantages and limitations of LSTM of increasing time complexity, incapable of keeping precise count of time step etc[17].

Another similar paper [18] analyzes the predictions, limitations, representations and error of recurrent neural network models. The authors described the three different models RNN, LSTM and GRU, one better and efficient than its predecessor. The drawbacks of RNN are backed up by LSTM and similarly GRU solves issues regarding LSTM. The paper used data-sets from Leo Tolstoy war and peace novel and source code of linux kernel which further were concatenated in a single file of 6,206,996 character long dataset. The authors also analysed the performance of RNN by comparing it with n-gram models that are a probabilistic language model for predicting the next item in a sequence. It is seen that for smaller datas both perform the same but for larger datas RNN outweigh with better performance. The paper used the identification of closing brace as a test case to check the ability of LSTM to work on different time frames as the range between open and close closing brace can vary from 10 to hundreds characters. The results suggest that LSTM can efficiently work up to 60 characters and performance degrades gradually. Error analysis between the models has also been performed and finally, concludes that networks can learn powerful and long-range real-world data[18].

Another very innovative work was done in paper [19] which also researched on how chatbots can be improved. The author of the paper stated how deep learning can play a significant role in developing realistic chatbots by "mimicking human characteristics". Initially, the author explained the different approaches such as knowledge based, Machine Learning and Deep Learning to build a chatbot. Knowledge-based has predefined responses which is further improved by ML that finds relation between patterns in text and the output value. Both the domains need to have a fixed data representation to make predictions and this is where deep learning comes into play. Deep learning helps to read the data the way it is and also can train large datasets at a time. Its complex structure helps to model the human mind and is able to make logical interpretations unlike the previous two models. By this paper, we were able to learn the benefits of working in the field of Deep Learning. The author further briefed about the different neural network models and implemented a hybrid chatbot using rule-based and generative models together to make a more meaningful response. The paper also showed comparison between conversations by using different Neural Network models like Uni-directional LSTM, Bi-directional LSTM, Bi-directional LSTM with attention mechanism. Bleu Score and Turing test scores were used to find the best model for the work. Moreover, the paper focused on extracting the context of the dialogue and thus the hybrid model gave better scores [19].

To enhance our knowledge on emotion detection we investigated the research paper [20] that gives us a significant understanding on emotion recognition: its importance and challenges. The author explains emotion recognition is a detailed version of sentimental analysis which only works on polarity where else ER aids to get in depth detection of the user's current state of mind and wellness. This can help businesses to get individual user's opinion on the purchase and through identification of emotion expressed in the review, organizations can work better to make improvements in their products and services. The paper further discusses the various datasets to perform emotion recognition and this helped us to get the desirable dataset for our implementation. In addition, it tabulated a list of already implemented emotion recognition and their limitations. This was extremely helpful as we were able to find our place to work with[20].

We also examined the paper [21] that detects emotion from a text using Bi-directional neural networks and tensorflow libraries. It describes how sentiment and emotion are interconnected and can be used to give better responses. The dataset is cleaned, processed and tokenized to fit to Bi-RNN model for training. The model had two parameters, one for sentimental analysis and another to generate dialogue. The work wasn't very efficient as they simply labeled the emotion as happy if sentiment is positive and sad if sentiment is negative. It did not classify the emotions and thus, failed to actually signify the improvement that could be achieved by classifying detailed emotion from the polarity of sentiments. The model played a happy song if the sentiment was positive and vice versa. However, our work completed the drawbacks of this paper by distributing emotion into different classes[21].

In addition, paper [22] surveyed the effectiveness of implementing Named-Entity Recognition by deep learning. The author gave the information of named-entity datasets and where they can be used. The paper stated the different evaluation metrics of NER such as type, exact match, relaxed F1 and strict F1 metrics etc to analyse the performance and accuracy of NER. The survey analysed knowledge-based systems, unsupervised systems, feature-engineered systems, feature-inference neural network systems, word level architectures, character-level architectures, character+word level architecture and lastly, character+word+affix model. After evaluation it was noticed that neural network models outperformed all other applied models mentioned above[22].

Furthermore, the paper [23] proposed their research on Dialogue intent classification by hierarchical LSTM which is an open source library. Through this a model will be able to recognise the reason behind the uttered sentence. The experiment was performed on a chinese ecommerce site. The authors compared accuracy between basic LSTM, HLSTM and their hybrid model HLSTM + Memory. The model outperforms others by 83% where else the others were around 79% and 81%[23].

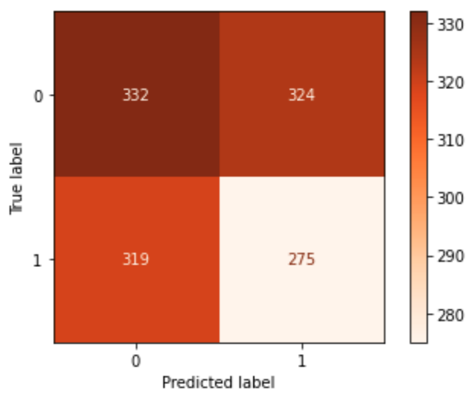
By going through all the knowledgeable and qualitative research, we gathered the loopholes of all the papers that needed improvement as well we ensured that our working domain, deep learning, is the right choice for incorporating all the works we studied so far. In our literature review, we have seen the development of models by sentimental analysis or emotion or intent or named-entity. However, so far our research there hasn't been yet an analysis that clusters all the components together. Thus, we decided to observe the different performances as well as challenges while using all the features together.

Chapter 3

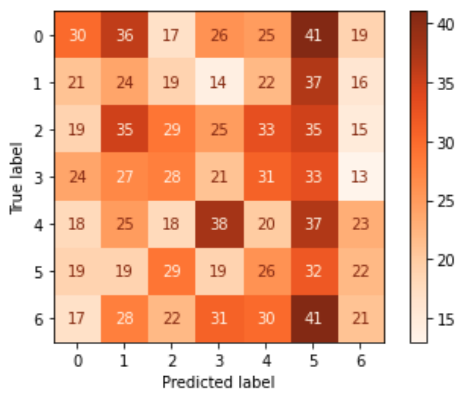
Model Selection

3.1 Background

Initially, we used a machine learning algorithm, Support Vector Machine, to evaluate performance on sentiment analysis and emotion detection. Due to its mathematical foundation in statistical learning theory, Support Vector Machines have become very popular in the field of machine learning and have been used for supervised classification[24]. SVM is a powerful and supervised algorithm for classification of vectors of real valued features and has been extensively used in classifying the polarity of a text[25]. The machine learning algorithm SVM can also be used to recognize, interpret and process human emotion from text according to predefined emotion classes[26]. Figure 3.1a and Figure 3.1b shows results from sentiment analysis and emotional detection with SVM were not very satisfactory. It gave only 48.5% accuracy for sentiment analysis and 14.16% for emotion detection. Thus, the poor performance of ML algorithms shifted us to the field of deep learning. Various deep learning techniques were applied for identification of sentiment, emotion, intent and entity from text which plays a great role in making the conversational agents more human-like and connected to the user.



(a) confusion matrix for sentiment analysis



(b) confusion matrix for emotion detection

Figure 3.1: Performance evaluation using Support Vector Machine

3.2 Deep Learning Model

3.2.1 Long-Short Term Memory (LSTM)

LSTM is one of the most advanced deep learning model that has the ability to learn long-term dependencies and learns patterns from previous states to generate predictions.LSTM achieves it by prioritising what things to remember and what things to discard.

To overcome longer dependencies the LSTM has 3 gates - 1. Input gate 2. Forget gate 3. Output gate

All these gates work to control and protect the memory cell states of LSTM which stores information.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3.1}$$

The first function is in the forget gate layer that determines which information to keep and which information to discard. It takes input h_{t-1}, previous state and x_t, current input and based on these gives an output result between 0 to 1. If the output is 0 it means delete the information and 1 means to keep the information.

$$i_t = \sigma(W_i.[h_{t-1},\, x_t] \;+\; b_i]) \tag{3.2}$$

$$\tilde{C}_t = \tanh\left(W_C.[h_{t-1},\, x_t] \;+\; b_C\right) \tag{3.3}$$

Next, we decide which information to keep. This happens in the input gate layer using the sigmoid function $i(t)$. Also, C_t stores the new candidate values that can be used in the upcoming state.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.4}$$

In this part the previous state C_{t-1} is multiplied with the ft to delete the forgetting information decided. It is then added to the product of new candidate values which is scaled to the values of input chosen.

$$o_t = \sigma(W_o\, [h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{3.6}$$

Finally, we use o_t to decide which parts of the output should be used to proceed. This o_t is multiplied with tanh function to give us our desired output.

Limitations of LSTM

1. It increases computational complexity.
2. Constant error flow can be used only in truncated LSTM.
3. All the gradient-based algorithm problems fail to precisely count the time steps.
4. LSTM is a multilayer-perceptron model which requires multiple repetition of each cell and they run for each sequence time-step. As a result the memory bandwidth to be computed is very large and the system fails to provide enough memory bandwidth to feed the computational units.
5. Random weight initializations have an effect on LSTM which makes it behave like a feed forwarding neural net.
6. Dropout is a regularization method which reduces overfitting and improves the performance of models. It becomes a difficult task to apply a dropout algorithm in LSTM as it is inclined to overfit.

3.2.2 Bi-directional LSTM (Bi-LSTM)

We know basic LSTM stores the history of the input words that passed the hidden layer, however, it only stores the words from the past as only those have been passed through it. This is a problem as to understand complex sentences we need to read the line from starting to end. Thus, Bi-directional LSTM helps us achieve that by concatenating two LSTM models together, one which reads a sentence from past to future and another from future to past. Bi-directional LSTM has two hidden layers that incorporate both the information gathered from two LSTMs and generates better results.

3.2.3 Gated Recurrent Unit (GRU)

To overcome the computational complexity of LSTM, we are using another model GRU which has less computational time and is more simpler and efficient than LSTM. GRU has two gates which are reset and updated where else LSTM has 3 gates. It is simple as it doesn't have a memory cell and directly dives to the hidden states. Moreover, its performance is almost similar to LSTM provided the less blocks of functions used.

$$r_t = \sigma(W_r \, . \, [h_{t-1}, \, x_t]) \tag{3.7}$$

This is the function of the reset gate in GRU which gives a value based on relevance to the previous state.The value ranges between 0 to 1 that decides its priority.If $r == 0$, that means the previous state should be forgotten.

$$\tilde{h_t} = \tanh\left(W.[r_t * h_{t-1},\ x_t]\right) \tag{3.8}$$

Using the relevance function and previous state we find the output of the current state.

$$z_t = \sigma(W_z.[h_{t-1},\ x_t]) \tag{3.9}$$

This is the update gate which decides which part of the updated candidate value will be used to compute the current state and it also decides the part of the previous state that is restored.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{3.10}$$

The value of z_t is multiplied with the updated candidate $\tilde{h_t}$ to refrain the scaled updated candidate value which is again added to the previous state times the value of part that needs to be selected from h_{t-1} state. Hence, this summation gives us the current state cell value.

Limitations of GRU

1. Their inherent sequential behavior makes them unfit for increasingly massive datasets and exponentially slower to train.

2. Slow convergence and low learning efficiency in GRU results in underfitting.

3.2.4 Bi-directional GRU (Bi-GRU)

Despite GRU being better than LSTM, it also lacks context from the future. Hence, Bi-directional GRU is better as it extends the basic GRU by adding another second layer to connect the hidden networks that flows in opposite directions to each other and thus captures context of both past and future at the same time. As it is a hybrid model of GRU and Bi-directional LSTM it tends to perform better than the individual models.

Chapter 4

Experimental Analysis and Results

4.1 Sentiment Analysis

As our goal is to create a human-like understanding of text by a chatbot, it is important to understand the sentiment of the content. Sentiment analysis is helpful to analyze the information in texts where opinions are highly unstructured and are either positive or negative[27]. Various machine learning algorithms like Support vector, Naive Bayes, Max Entropy are applied to perform sentiment analysis [27]. For humans, it is very simple to identify positive and negative sentences. But, a chatbot can never understand the polarity of a text if it is not trained to identify it. Sentiment Analysis is very essential for any chatbot to capture whether the user is saying something positive or negative. Without understanding the sentiment of the user, the chatbot will not be able to give appropriate responses to the user. So, here, we are proposing different deep learning models to analyse the polarity or valence in text. We applied the simple neural network, GRU and LSTM to perform sentiment analysis on two different datasets.

4.1.1 Data Collection

For our work, we used two datasets for analysing our performance in different neural network models with different datasets. In Table 4.1, the first dataset is described which is IMDB dataset of movie review consisting of two columns text and sentiment[28]. The text column is the review on movies and the second column is the sentiment column that gives the output of *"positive"* or *"negative"* review. In Table 4.2, the second dataset twitter sentiment analysis dataset is described which also has two columns as well[29]. The text column gives various sentimental twitter comments and the sentiment column gives the binary result that is '0' for negative sentiment and '1' for positive sentiment.

4.1.2 Data Processing

Firstly, we have plotted the sentiment of the two dataset by splitting the positive and negative sentiment and checked the ratio of the number of sentences with the two sentiments. In Figure 4.1a, the sentiment ratio of IMDB dataset is shown and in

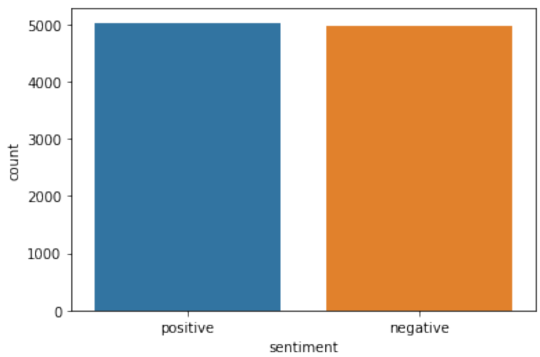
id	text	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production...	positive
2	I thought this was a wonderful way to spend ti..	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Table 4.1: IMDB Dataset for Sentiment Analysis

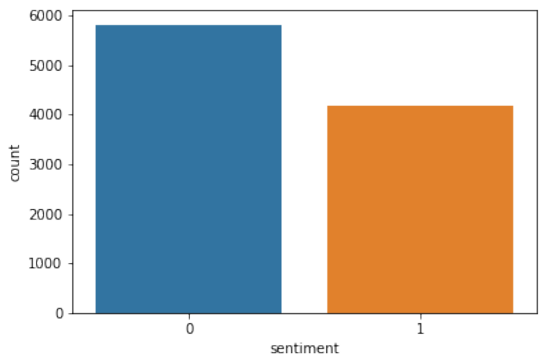
id	text	sentiment
0	is so sad for my APL frie...	0
1	I missed the New Moon trail...	0
2	omg its already 7:30 :O	1
3	.. Omgaga. Im sooo im gunna CRy. I'...	0
4	i think mi bf is cheating on me!!! ...	0

Table 4.2: Twitter Sentiment Dataset for Sentiment Analysis

Figure 4.1b, the sentiment ratio of twitter dataset is shown. The positive sentiment is described in blue and negative sentiment described in orange.



(a) Ratio of sentiment for IMDB Dataset



(b) Ratio of sentiment for Twitter Sentiment Dataset

Figure 4.1: Ratio of sentiment for Sentiment Analysis Dataset

In Figure 4.2, the whole workflow process of sentiment analysis is described in depth. The first dataset is cleaned by removing html tags, multiple spaces, meaningless words, single letters, converted all n't to not, etc. We converted the dataset to numpy-array and the *"positive"* and *"negative"* sentiment to '0' and '1' so that both the datasets have similar forms of output.

The second is cleaned by removing meaningless words, hyperlinks, mentioned names using @, words starting with &, multiple dots, slashes & hyphens, single letters and also converted all n't to not. Dataset 2 is also converted to a numpy array and checked both the datasets after refining.

Afterwards, we splitted all two datasets to test and train sets in the ratio 1:3 respectively. Then, we tokenized the datasets separately to fit into texts and further converted texts into sequence. For ensuring all the sentences are of equal length, we padded the tokenized sequence to max length of 100.

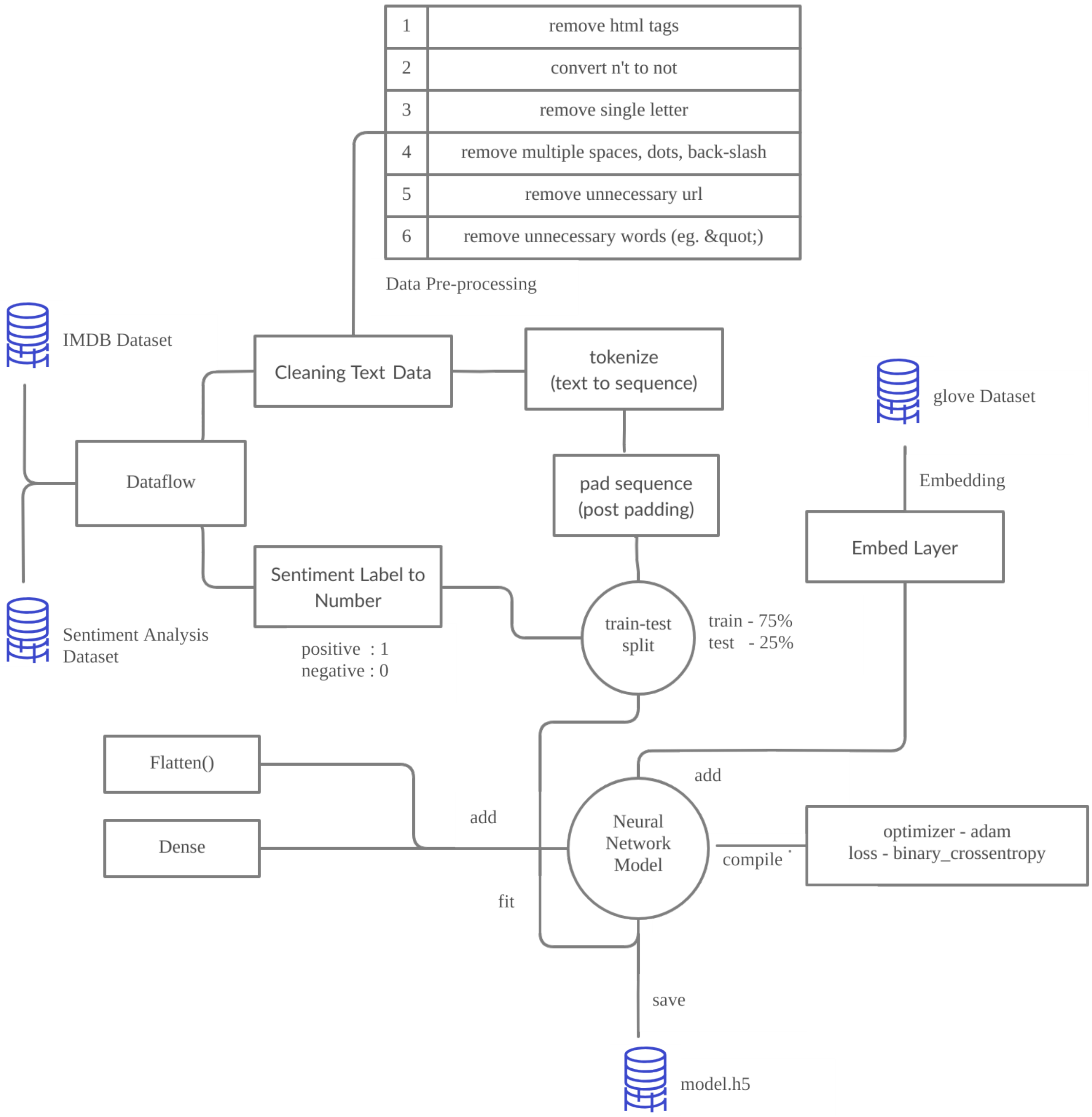


Figure 4.2: Training Work Flow for Sentiment Analysis

4.1.3 Method

We can feed our textual data into a model to train. The text has to be converted to numerical forms so that we can feed them into the model. The conversion of text into vector representation is done by the help of word embedding.

For capturing the context of a word in the datasets, we used Stanford's Glove word embedding which is an unsupervised learning algorithm for obtaining vector representations for words[30]. The benefit of using GloVe is that it focuses on global statistics that is word co-occurrence rather than depending on the local context information of words[43]. We can get an idea about the semantic relationship between the words through GloVe.

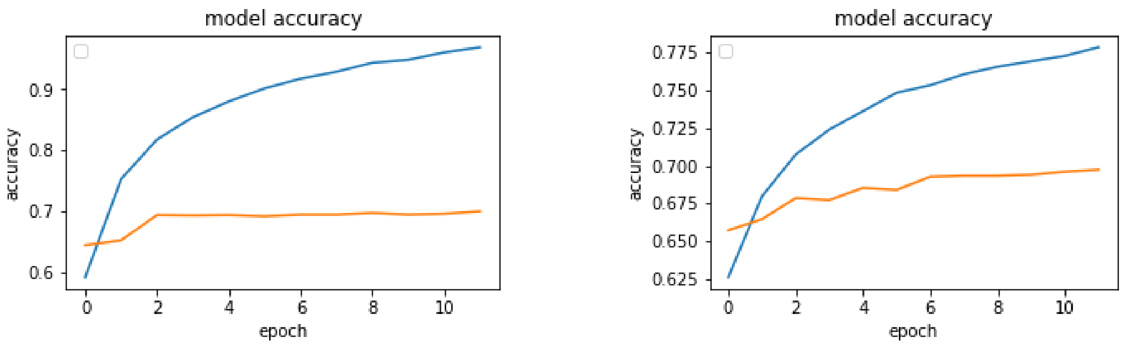
As different neural network models give different results and accuracy we tried to implement a number of them such as Simple Neural Network, LSTM and GRU. The output of any neural network is determined by activation function. In our models we have used sigmoid activation function as it can convert the models output into a probability so it becomes easier to work with[44]. It also maps the value to the range of 0 to 1 and is also helpful to update or forget data in LSTM as the closer to 0 means forget and closer to 1 means keep. For reducing the losses and to provide the most accurate possible results optimization algorithms are used. We have used Adam optimizer because it is a stochastic gradient descent method that is based on adaptive estimation. The estimations of first and second moments of gradient is used to adapt the learning rate for each weight. Since, it is a binary classifier we are using binary cross entropy as our loss function. The main reason behind using loss function is that it should return high values for bad predictions and low values for good prediction. Cross entropy is the measure of difference between two probability distributions for a random variable. It is built upon the idea of entropy which is the number of bits required to transmit a randomly selected event. Sigmoid activation function is pretty compatible with binary cross entropy.The number of examples from the training dataset used in the estimation of error gradient is the batch size. It was important to keep our batch size moderate to get rid of the overfitting problem and control the accuracy of the estimate of the error gradient while training neural networks. So, we kept the batch size at 128 for all our models. Each time the algorithm has seen all the samples in the dataset, one epoch has completed. The epoch was set to 20 means the algorithm will see the entire dataset 20 times. Validation split is used to determine the loss and any model metrics on this data at the end of each epoch.The validation data is selected from the last samples in the x and y data provided before randomizing.The remaining fraction of the data is trained by the model.We kept the validation split to 0.2. After we feed the two datasets into the model following the above parameters, we are going to show the different accuracy results of the two datasets.

4.1.4 Experimental Results

In Table 4.3, we showed comparison between the accuracy and loss of different DL models. From our performance analysis we found LSTM gave the highest accuracy in both test and train dataset. The train accuracy in dataset 1 and 2 are 89% and 86% and test accuracy are 80% and 71% respectively. In other models the test accuracy dropped significantly compared to training accuracy. We plotted the accuracy and loss graph in Figure 4.3 and Figure 4.4 for simple neural networks, Figure 4.5 and Figure 4.6 for LSTM and lastly, Figure 4.7 and Figure 4.8 for GRU. For each model we showed comparison in results between IMDB and Twitter dataset. The epoch was plotted along the x-axis and accuracy was plotted along the y-axis. The train accuracy is represented by the blue line and the orange line represents the test accuracy. From graphs, we can see GRU outperforms all other models in case of training, however, degrades in testing.This is due to overfitting problem that commonly happens in most neural network models while training. LSTM works comparatively well in both train and test phases.

model	dataset	train data		test data	
		loss	acc	loss	acc
Simple Neural Network	IMDB Dataset	0.37	0.86	0.82	0.68
	Twitter Sentiment	0.54	0.74	0.66	0.65
Gated Recurrent Unit	IMDB Dataset	0.32	0.91	0.95	0.79
	Twitter Sentiment	0.74	0.80	1.19	0.68
Long-Short Term Memory	IMDB Dataset	0.33	0.89	0.65	0.80
	Twitter Sentiment	0.43	0.86	0.92	0.71

Table 4.3: Accuracy for training and validation data of different Datasets and different Deep-Learning Models for Sentiment Analysis

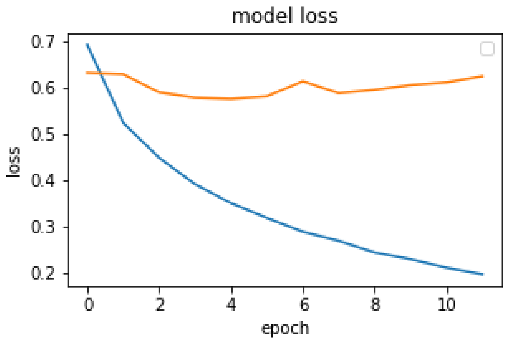


(a) IMDB Dataset (b) Twitter Dataset

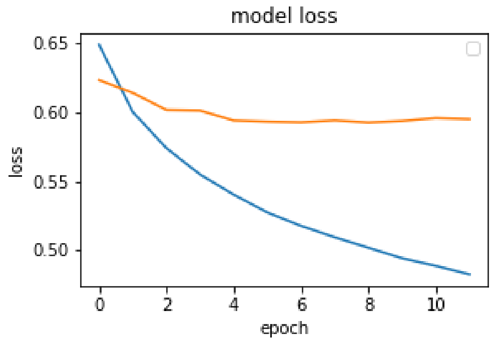
Figure 4.3: Accuracy graph for Simple Neural Network

4.1.5 Performance Analysis

Finally, in Figure 4.9, we gave some sentences as input to make some predictions whether the sentences were classified correctly as positive or negative. We have set the range of confidence level of a sentence from 0 to 1. The sentences which have

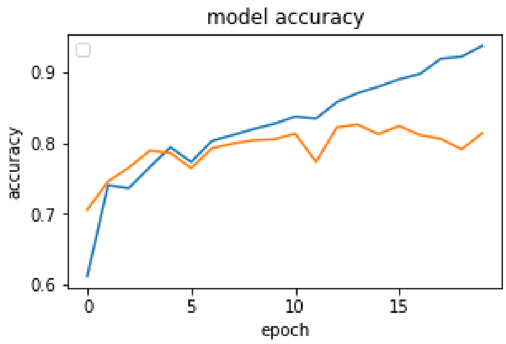


(a) IMDB Dataset

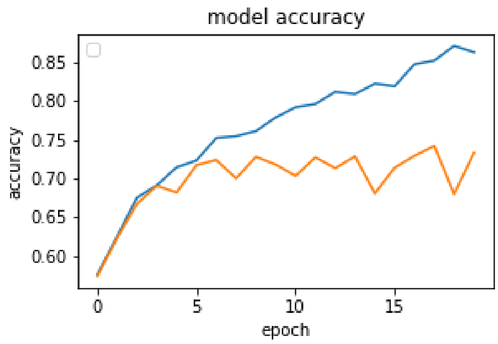


(b) Twitter Dataset

Figure 4.4: Loss graph for Simple Neural Network

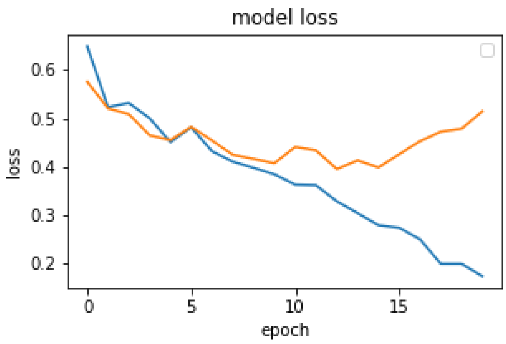


(a) IMDB Dataset

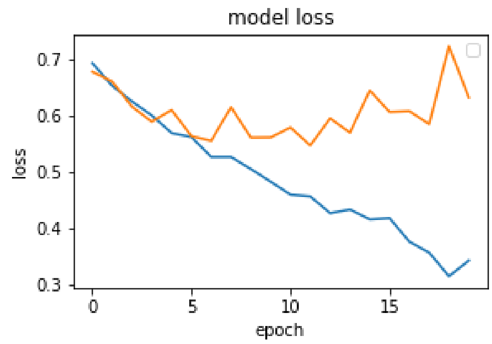


(b) Twitter Dataset

Figure 4.5: Accuracy graph for LSTM



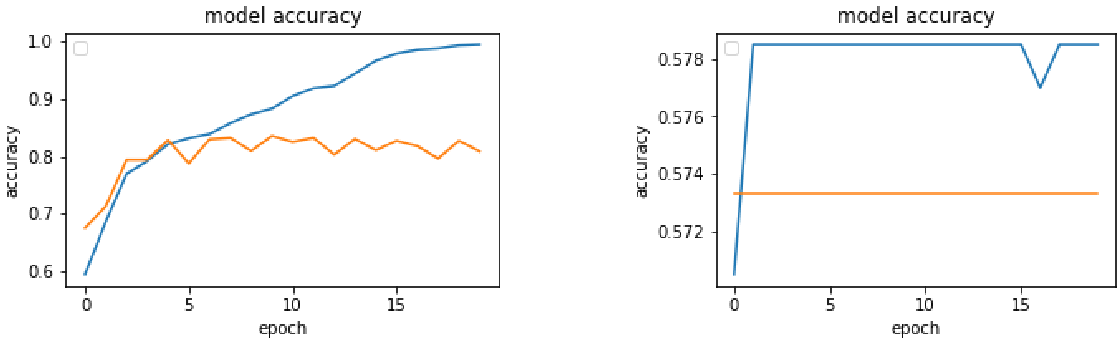
(a) IMDB Dataset



(b) Twitter Dataset

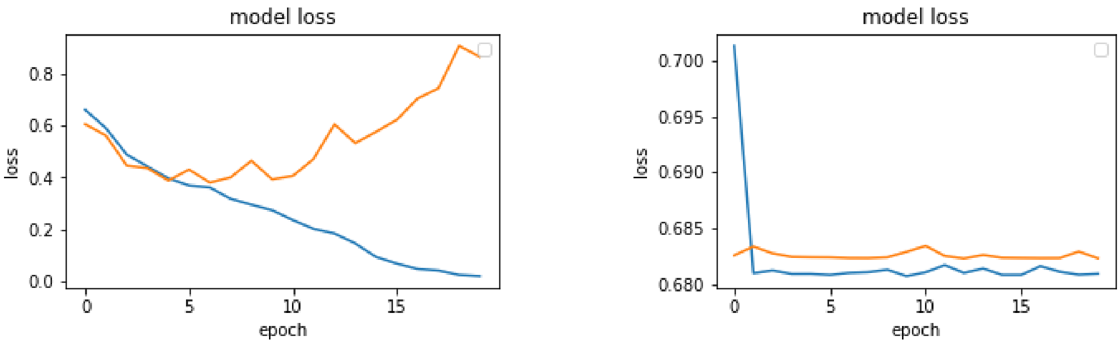
Figure 4.6: Loss graph for LSTM

confidence level greater than or equal to 0.5 are given a positive sentiment label. If it is less than 0.5, a negative sentiment label is given. Initially, we gave simple sentences as input and the bot was able to predict the accurate sentiment. As our bot is trained with different sentences that have words and patterns of positive and negative polarity,the bot identifies those structures and vocabularies such as good, bad etc. to predict the sentiment.Usually, it is easier to extract sentiments from sim-



(a) IMDB Dataset (b) Twitter Dataset

Figure 4.7: Accuracy graph for GRU



(a) IMDB Dataset (b) Twitter Dataset

Figure 4.8: Loss graph for GRU

ple sentences but when the input text is longer with multiple expressed opinions the machine, in most cases, fails to give correct sentiment. However, our third example was a complex sentence which contains mixed judgement. By our knowledge we are able to understand that this particular sentence focuses on the negative portion and we can see that even our bot is able to catch the right sentiment despite the complexity of text .Moreover,our bot is able to predict from consecutive sentences given in the same input. Thus, we can say our bot is performing well in sentimental analysis.

4.1.6 Sentiment to Emotion transition

From our analysis, we can say that LSTM is the best model for Sentiment analysis. By now we have only reached to learn the attitude of a person from a user's input. This gives us a general idea of the polarity of thoughts the person is going through. For example- *"What a dream it was! I couldn't sleep at all."*, here by sentimental analysis we can classify this text as a negative sentence, however, this is not enough to figure out what specific kind of negative opinion it is. To understand a person's situation, we need to identify the exact mental state of the user. Let's analyse the above example again. If we closely analyse the text we can see the person uttered

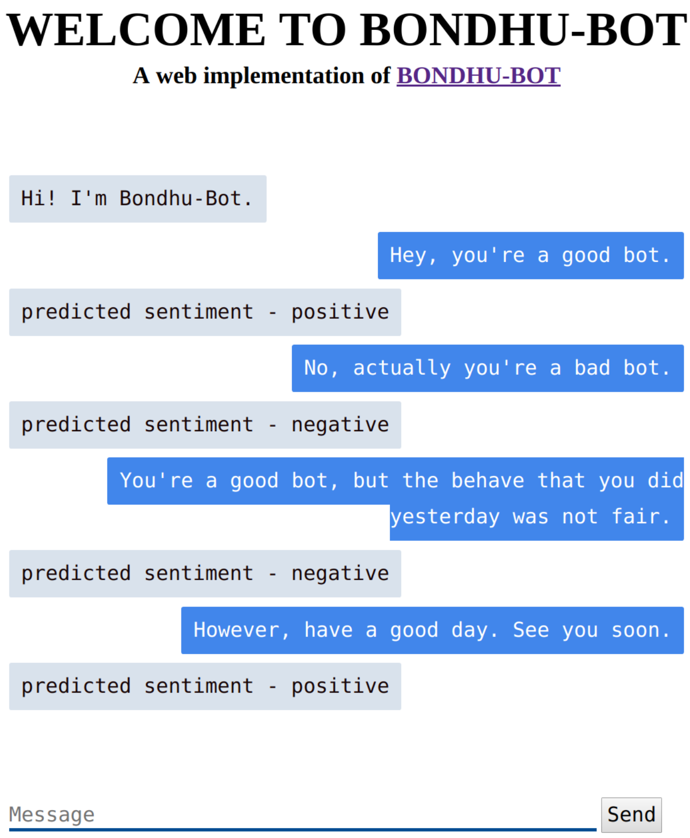


Figure 4.9: Prediction for Sentiment Analysis

the sentence out of some *“sadness’* and some *“fear”*. This is the exact state of mind we need to capture in our chatbot. Hence, one possible solution could be to recognise the expressed emotion in the text by emotion analysis[31].

4.2 Emotion Detection

Emotion Detection will play a promising role in the field of Artificial Intelligence, especially in the case of Human-Machine Interface development[32]. Emotions are very subtle and ambiguous in a text which makes it really tough to detect. For that reason, we have focused on using neural network models in our research to capture the emotion of the user. The representation of emotions can be explained using emotion models. Among the three branches of emotion models, categorical is used for the research of textual emotion detection as it divides emotions into several groups and assigns the emotion into proper discrete categories[33]. The ekman model differentiates between six different categories of emotion like anger, disgust, fear, joy, sadness and surprise[34]. The plutchik model which is similar to Ekman, proposes that there exist few primary emotions, which occur in opposite pairs and named eight of such fundamental emotions, that is, acceptance/trust and anticipation in addition to the six primary emotions in Ekman[20]. According to paper [35], Long short term memory (LSTM) gives better results in detecting a user’s emotion compared to

convolutional neural networks (CNN) along with word embeddings. In our proposed model, we develop Bi-directional GRU and Bi-directional LSTM as deep learning models for more accurate results in emotion recognition.
We used the following steps for emotion detection:

4.2.1 Data Collection

The dataset used in this model is International Survey on Emotion Antecedents and Reactions (ISEAR) database created by Klaus R. Scherer and Harald Wallbott which consists of 7665 sentences labelled by seven emotions like joy, sadness, fear, anger, guilt, disgust, and shame[36]. The dataset contains two columns: emotion label and text. Figure 4.10 classifies the different categories of emotion and the proportion of data we have for each of them.

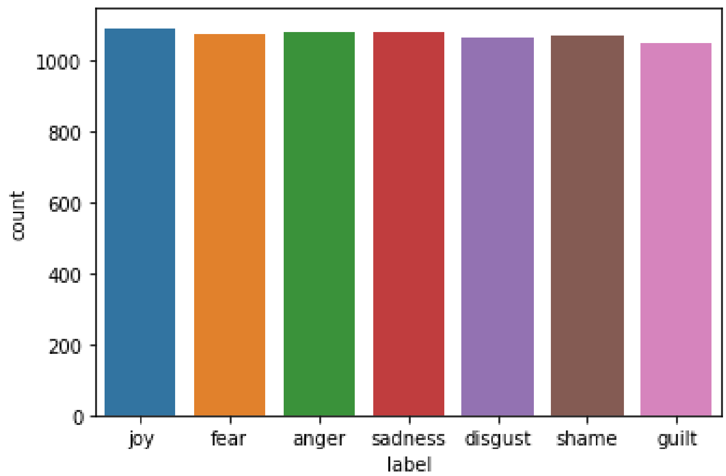


Figure 4.10: Ratio of different emotion in ISEAR Dataset

4.2.2 Data processing

As shown in Figure 4.2, we cleaned the data removing stopwords and tags and splitted all the phrases into smaller parts called tokens using Keras tokenizer. Then, we have converted them to sequences and padded them to maximum length. Padding the input sentences is very necessary as all the neural networks require having inputs of the same shape and size. Words can be in different forms according to their grammatical function in a sentence. All these forms of a single word are clustered by a process called stemming. Stemming is an important part in the natural language understanding(NLU)[45]. It is the process of reducing a word stem to its root known as lemma and used to extract information[45]. For stemming we used the Lancaster stemmer of nltk (Natural language toolkit). We focused on onehotencoder as we are dealing with categorical data in emotional analysis. Categorical data contain different labels and some categories are related to each other. But for training the model we need to convert the categorical data into numerical form which is done by One-Hot encoding. So, we used One-Hot encoder from scikit library. Finally, we splitted the test and train data by keeping the test size to 0.2.

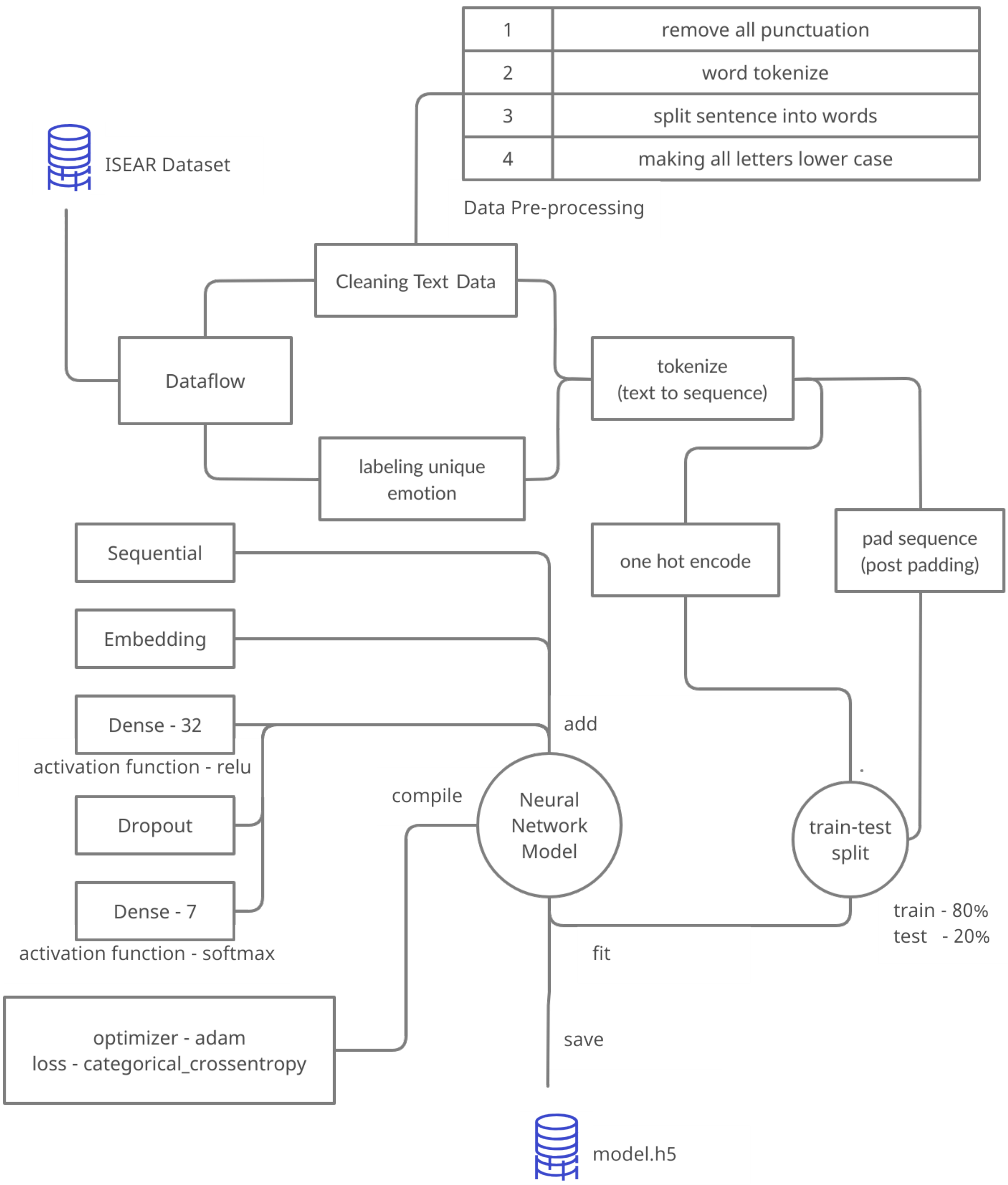


Figure 4.11: Training Work Flow for Emotion Detection

4.2.3 Method

For the analysis we have used Bi-GRU and Bi-LSTM and did some comparison which works best for our analysis. Again, we took the help of Keras library for creating sequential models. We used ReLU (Rectified Linear Unit) and softmax as our activation function. The activation function which is a piecewise function that will output the input directly if positive or else it will output zero is the rectified linear activation function(ReLU)[41]. We have used both softmax and ReLU so that it can convert the unbounded input into an output that has predictable form. Simulation of having large number of different architectures can be done by a single model by randomly dropping out nodes during training which is the mechanism of dropout in deep neural network models[42]. The dropout was set to 0.5 for effective regularization, to reduce overfitting and improve generalization error in the models. We used categorical cross entropy as our loss function as we are dealing with categorical data. Finally, we have set the optimizer to Adam for accurate results. Moreover, for saving the model so that it can be loaded later to continue we used the model checkpoint callbacks of Keras callback library. The liveloss plot tool was applied so that we could get the training loss plot. The batch size is used to control the number of samples to work through before updating the internal parameters of the model. It is a recurrent calculation on a particular part of the data to save time and computational resources. We need to use a moderate size of batch so that it does not overfit the memory. For that reason we set the batch size to 32. The number epoch determines the number of times the algorithm sees the entire data set. So, each time the algorithm has seen all samples in the dataset, one epoch has completed. That is why epoch was kept at 100 for both the models so that the algorithm sees the entire dataset 100 times.

4.2.4 Experimental Results

Table 4.4 shows the accuracy and loss results between Bi-LSTM and Bi-GRU.Comparing the results,we can say Bi-GRU gives the highest accuracy of 91%.We plotted accuracy and loss graph in Figure 4.12 for Bi-LSTM and Figure 4.13 for Bi-GRU.From the graphs mentioned above we can see the performance of both the models were neck to neck.

model	training		validation	
	loss	acc	loss	acc
Bi-directional LSTM	0.227	0.908	2.204	0.684
Bi-directional GRU	0.325	0.910	2.203	0.919

Table 4.4: Accuracy for training and validation data of different Deep-Learning Models for Emotion Detection

4.2.5 Performance Analysis

In Table 4.5, we showed the background values of how an input sentence is classified on the basis of confidence level for each emotion. The emotion with the highest confidence level will be the output. Moreover, few examples with corresponding outcomes are given in Figure 4.14. Previously, our bot was able to predict sentiment only but now we incorporated emotion along with sentiment. Only sentimental

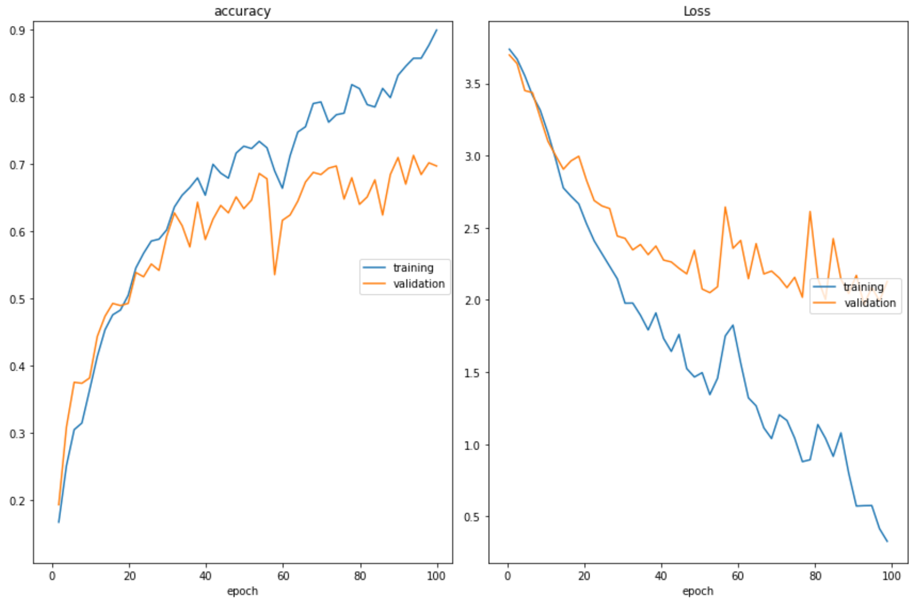


Figure 4.12: Accuracy and Loss for Emotion Detection using Bi-LSTM

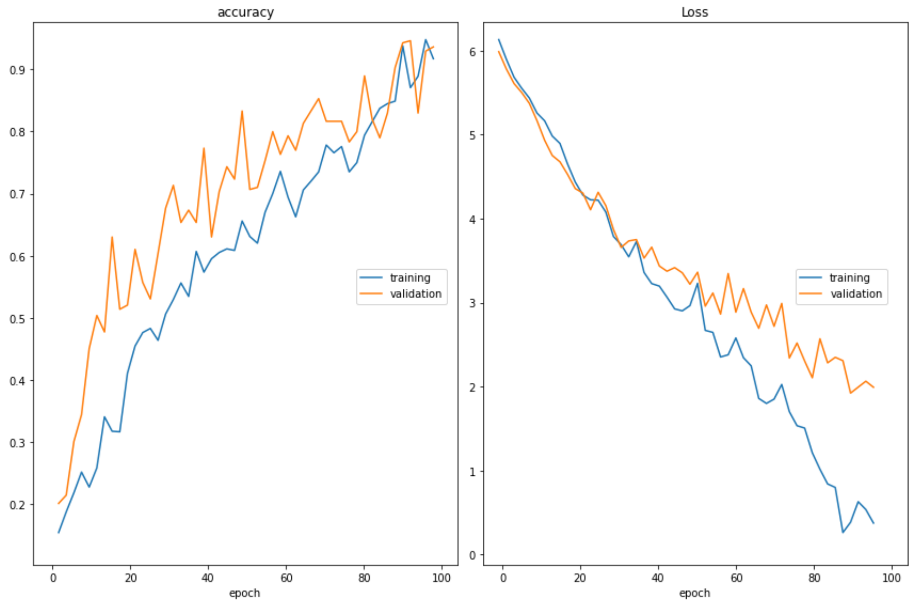


Figure 4.13: Accuracy and Loss for Emotion Detection using Bi-GRU

analysis can often be misleading as *"negative"* doesn't mean the person is always sad. Disgust, guilt, shame, fear, anger all refer to negative emotions. From our analysis we can see sentences 1, 2, 4 and 5 all have negative sentiment, however, we get four different emotions out of them: guilt, anger, sadness and disgust. This is the specific state of mind of a user we wanted to capture through our bot. We even tested with an incomplete sentence *"when someone stole my bike"* and the result is *"anger"* which is quite close to how a human would naturally react when something gets stolen. Emotion predictions for complex sentences are also achieved by our bot. In addition we can see that the bot can predict accurately without even using any of the seven *"emotion category"* words we have. Therefore, our bot is qualified to evaluate human reactions and mental state.

I did not help my thesis team enough.	
emotion	**confidence**
guilt	0.49738222
shame	0.36354083
anger	0.05595107
sadness	0.037771154
fear	0.022115987
disgust	0.015596252
joy	0.0076424023
predicted result	**guilt**

Table 4.5: Predicted confidence level of different emotions

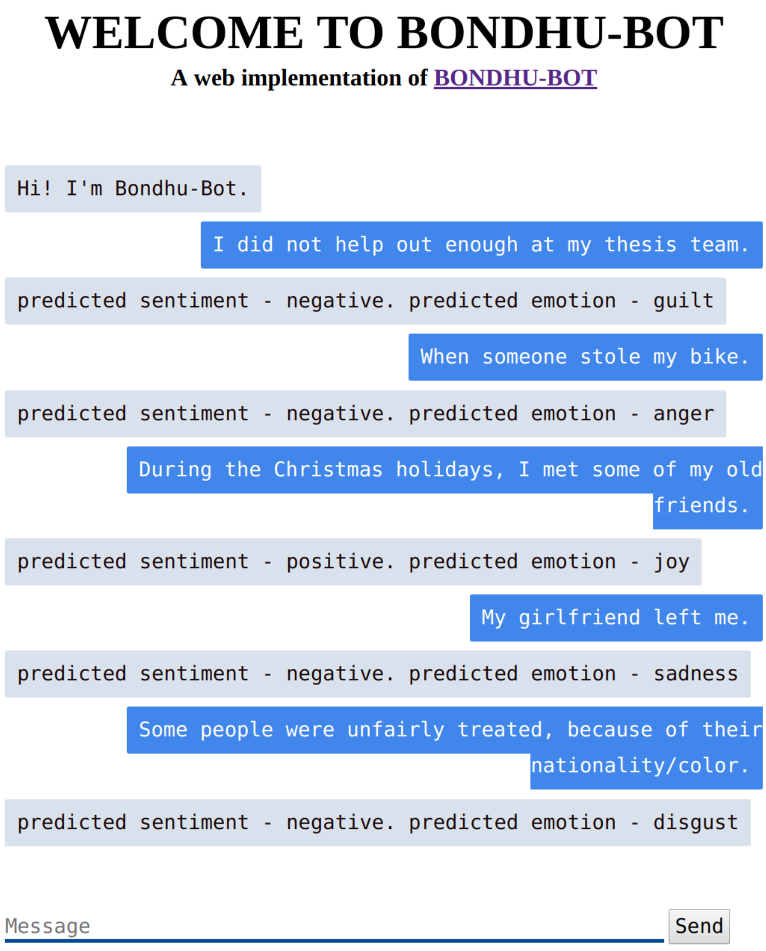


Figure 4.14: Prediction for Emotion Detection

4.2.6 Emotion to Intent transition

Though we got very accurate results through emotional analysis using Bi-directional GRU and Bi-directional LSTM, but for making the chatbot responses more human-like, knowing the intent of the message is essential. Intent analysis is basically understanding the intentions behind a text or speech. For understanding a person's mental health it is more important to know the reason behind the given input text and the associated emotion expressed. By emotion detection we will get the state of mind of the user, but *"why is he in that particular state?"* will be identified by knowing the intent behind it. And once the bot can detect the problem of the user, it can generate a suitable response accordingly. Suppose, the user sent a text message to the chatbot saying *"Good Morning! How are you doing?"* The intent of the message will be classified by NLU (Natural Language Understanding) component. With the help of this intent classification, an appropriate response will be generated by the NLG (Natural Language Generation), components like *"I am good. How* *about you?"* Besides, knowing the context of a person's text helps to gather various information. For example, if we keep on storing the history of the context of the person's text we can create an overall summary of an individual's personality. It helps knowing the person better.

4.3 Intent Classification

Intent Classification is very essential and the key to human machine dialogue systems[37]. To understand the user's utterance intent classification is necessary and proposed a deep hierarchical LSTM to classify the intent[23]. As conversational agents are being used for various purposes like customer support, e-commerce and even in the field of mental health, determining the intention of a user has become very essential. The chatbot will be able to serve the user more interactively if it can understand the intent of the user. In our proposed model, we have applied Bi-LSTM and Bi-GRU for the intent classification for making the chatbot more human-like.

We used the following steps for intent classification:

4.3.1 Data Collection

Table 4.6 lists the dataset banking-77, we used, which is composed of online banking queries and 78 corresponding intents[38]. It has two columns: sentences and corresponding intent of the sentences. As the sentences contain various capitalization, punctuations and special characters, we firstly removed them and refined our dataset to be usable to feed into our model.

id	text	intent
0	I am still waiting on my card?	card_arrival
1	How do I add a new card?	card_linking
2	What is an exchange rate?	exchange_rate
3	An extra fee of €1 was in my statement	extra_charge_on_statement
4	What currencies can I exchange?	fiat_currency_support

Table 4.6: Dataset for Intent Classification

4.3.2 Data Processing

The data is cleaned by removing all the punctuations and stopwords. The letters which were uppercase are made lower case for better training The sentences are splitted into words and further the words are tokenize using Keras Tokenizer. The stemming process was the same as emotional analysis. lemmatization groups all the similar words such as *"happy'*, *"happier'*, *"happiest'* to get the ultimate source word even if it is written differently. The lemmatization algorithm gathers all inflected forms of words so that they are broken down into their root form or lemma and also into different parts of speech[45]. We have done input text encoding by tokenzing into a sequence of words and giving indexes to each sentence as well as padded to a fixed max length so that they are equally in length and fit well. We have done output text encoding by tokenizing into indexes similar to the input text and then used one-hot encoder to fit the vector to the model. Like before the One-Hot encoder was used to convert the categorical into numerical form. Similar tools used for One-hot encoder in emotion detection were also used in intent classification. Figure 4.15 describes the steps we followed for data processing.

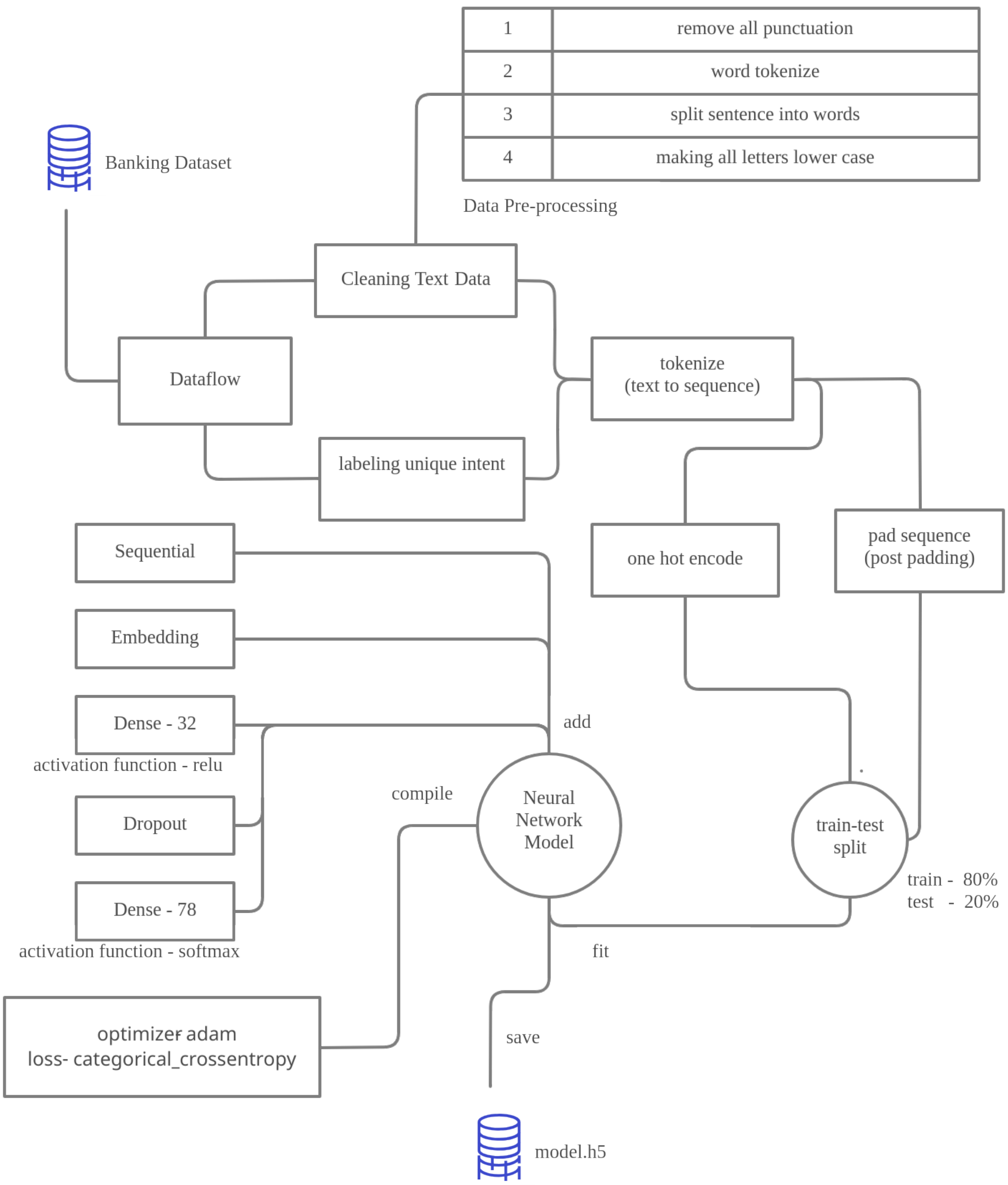


Figure 4.15: Training Work Flow for Intent Classification

4.3.3 Method

We splitted our dataset into 80% of training dataset and 20% validation. Now, our data is ready to be fed in the model. We used Bi-GRU and Bi-LSTM, as it is bidirectional. It creates a reverse copy of the sentences and reads from both the sides to keep both past and future context of sentence into consideration. Optimizer was set to Adam just like our previous anlysis and loss function to categorical cross entropy .The main reason behind using loss function is that it will return high values for bad predictions and low values for good prediction. Softmax and ReLU were used as activation functions and it is used the same way it was used in emotion detection and sentiment analysis. We used 100 epochs and batch size 32 as adjusting epochs and batch size properly stabilizes the accuracy of the model. All the tools used in the classification of intent were similar to the emotion detection and sentiment analysis.

4.3.4 Experimental Results

From Table 4.7 we got the training accuracy of around 80.7% using Bi-GRU and 77.4% accuracy using Bi-LSTM which indicates a good comparison between the two models.However, Bi-GRU gives comparatively better results for intent classification also .Our results are demonstrated in Figure 4.16 and Figure 4.17 as accuracy and loss graphs for Bi-LSTM and Bi-GRU.

model	training		validation	
	loss	acc	loss	acc
Bi-directional LSTM	0.651	0.774	1.337	0.711
Bi-directional GRU	0.562	0.807	1.468	0.694

Table 4.7: Accuracy for training and validation data of different Deep-Learning Models for Intent Classification

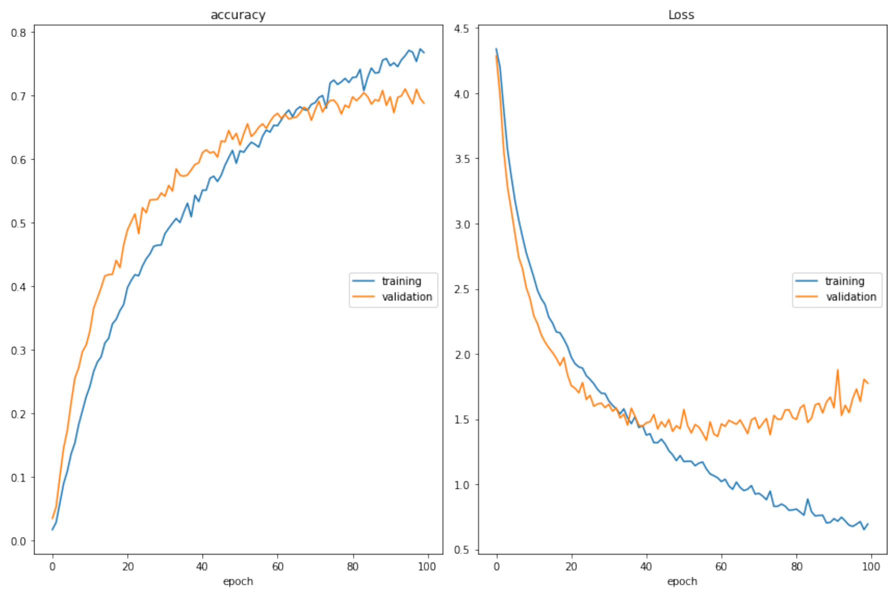


Figure 4.16: Accuracy and Loss for Intent Classification using Bi-LSTM

4.3.5 Performance Analysis

Just as before, we gave some sentences as input for real time prediction.We created a get_intent class that gives the possible probabilities of similarities of the sentence with the intent categories. The intent that gives the max probability is the resultant outcome.As our intent dataset is related to bank queries, our input sentences in Figure 4.18 are different statements or questions regarding bank issues. Despite having different datasets of sentiment, emotion and intent which were trained individually, we were able to link all the models to extract all three features.We used a single

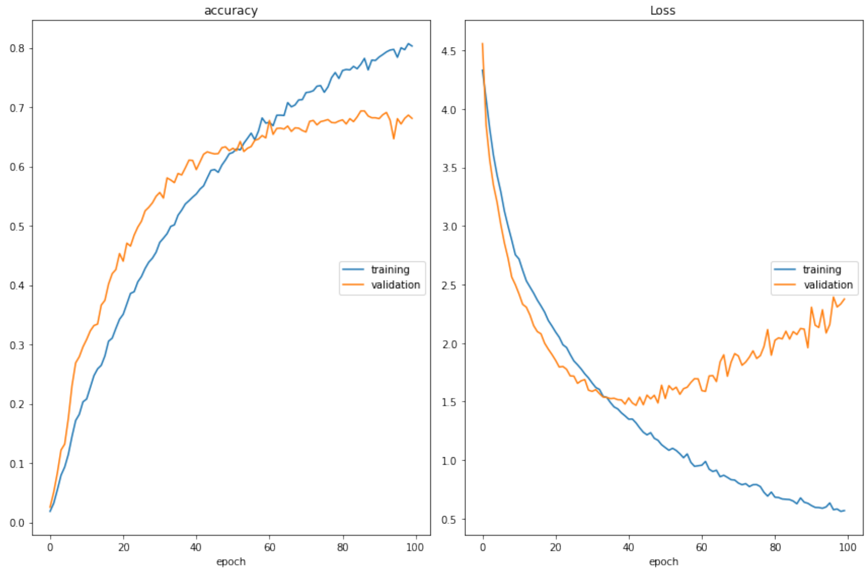


Figure 4.17: Accuracy and Loss for Intent Classification using Bi-GRU

class that sends the input text to all the method models and one single prediction parameter generates all the results from individual models.In our first example we can see the user's card isn't working and our bot predicted *"card_not_working"* correctly. Similarly,the intents of the following examples were identified without any mistake along with appropriate sentiment and emotion as well.

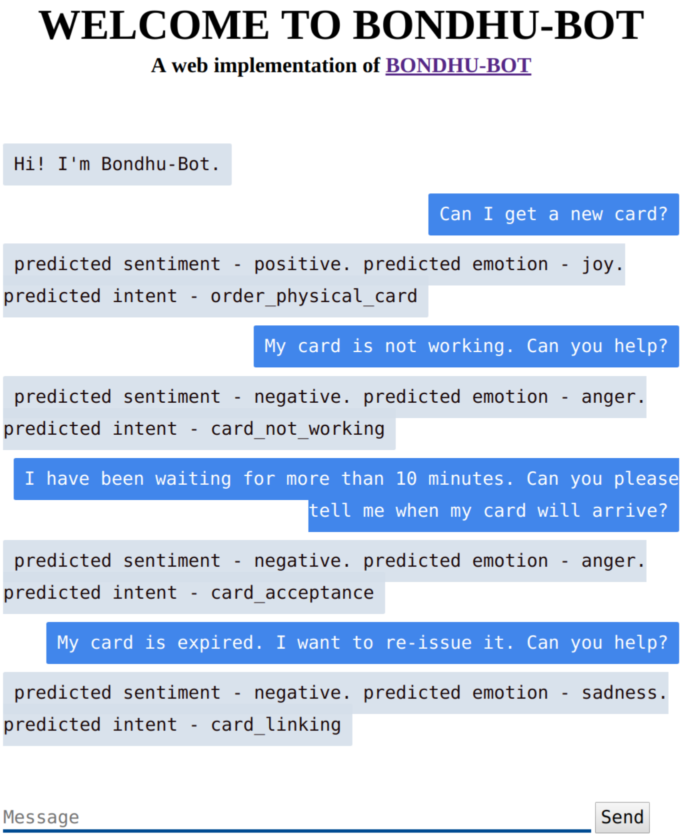


Figure 4.18: Prediction for Intent Classification

4.4 Named-Entity Recognition

Now, our next work is to get details of the intent. By *"details'* we mean about whom or where or at which date and time the context of the sentence is referring to. It is important because entities such as person, location, organization, time etc help to gather information related to the intent of the text. This kind of task is achieved by named entity recognition. NER is a subtask in information extraction and machine translation and also various DRNN models along with word embedding are applied to perform NER[39]. With the help of NER, we can identify and categorize key entities in text which will help our chatbots to become more interactive. Whenever humans communicate, we can easily understand who the person is, the person's name, the place the person lives. For instance, *"Alex loves to go to Dubai on vacation"*. Here, the chatbot will have to identify the entities Alex and Dubai and then categorize them to the person's name and location. But for a chatbot to identify these certain entities is a very challenging task. For helping the chatbot to recognize entities we need a named entity recognition system. This paper proposes a Bi-LSTM and a Bi-GRU architecture to perform named-entity recognition.

The following steps were followed to perform NER:

4.4.1 Data Collection

The dataset we used is Annotated Corpus for NER using GMB(Groningen Meaning Bank) corpus which is tagged and built for predicting entities such as name location, time etc[40]. We have assigned each token to either the *'O - category'* which means the word is not an entity or we have assigned them to one of the following entity shown in Table 4.8.

tags	full form
geo	Geographical Entity
org	Organization
tim	Time indicator
per	Person
gpe	Geopolitical Entity
art	Artifact
eve	Event
nat	Natural Phenomenon

Table 4.8: Different categories of entities for NER Dataset

4.4.2 Data Processing

Next, for processing the data, we have retrieved sentences and corresponding tags. As displayed in Figure 4.19, mappings between sentences and tags were defined by converting the words and tags to indexes. Next we have padded the input sentences which is the process of adding zeros at the end of the input sentences to make the samples in the same size. We have considered the maximum length as 50 which defines the maximum number of words in a sentence. After padding, the tag indexes were expressed as categorical.

4.4.3 Method

Now, we have created train and test splits so that we can estimate the performance of algorithms when they are used to make predictions on data. The training data was split to 90% and test data was split to 10%. Our next task was to build and compile bidirectional-LSTM. We are using bidirectional-LSTM, because our model can learn from the entire input rather than just the previous timestamps. We have applied Spatial dropout 1D to 0.1, is an alternative way of drop out. We specifically used Spatial dropout 1D to drop the entire 1D feature map. We also have kept the dense to 100 in our model. For training the model properly , we have used the ModelCheckpoint and livelossplot. Checkpointing the neural network model is essential as checkpoints are the weight of the model as these weights can be used to make better predictions.The batch size is set to 32 and epoch set to 3 just like our previous analyses. Finally we have done the training and evaluation of the model.

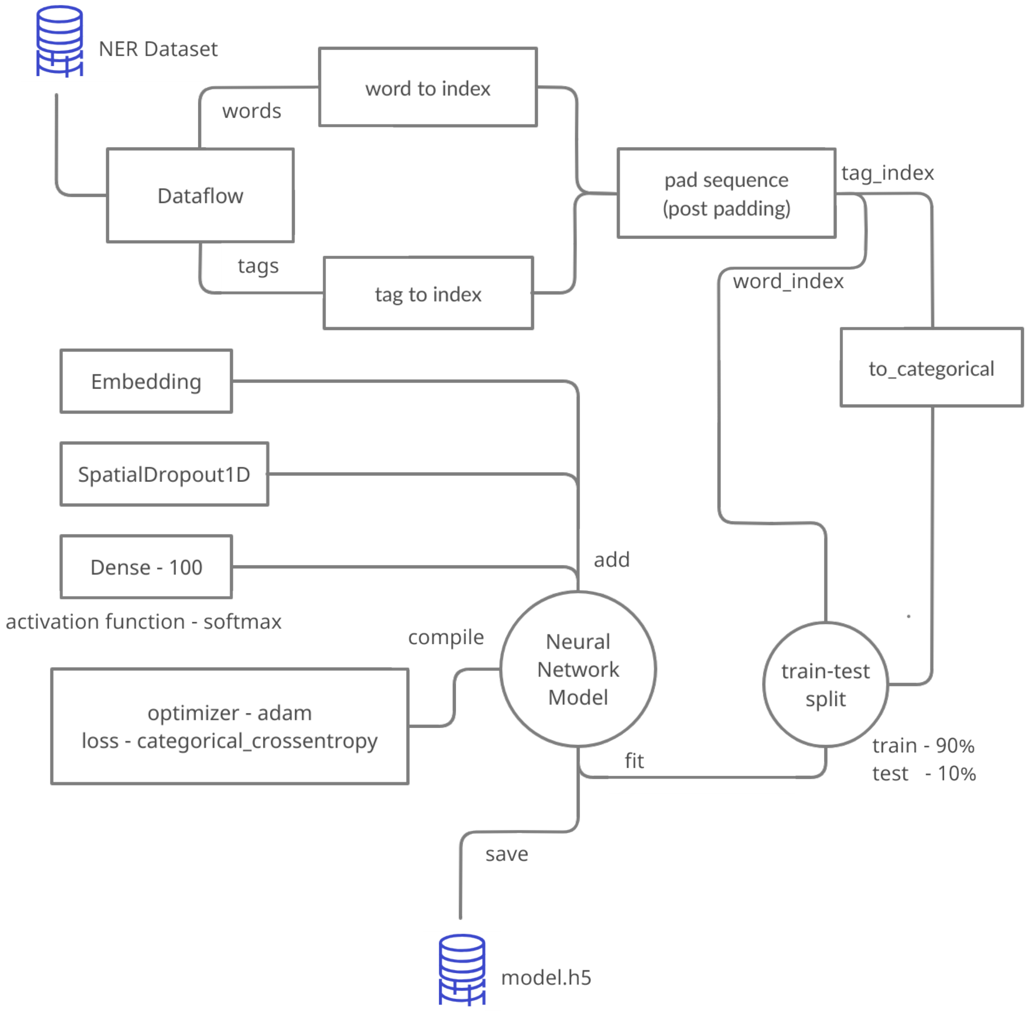


Figure 4.19: Training Work Flow for Named-Entity Recognition

4.4.4 Experimental Results

After training the model, we got an accuracy of 98.9% using both Bi-LSTM and Bi-GRU, given in Table 4.9,which plays a great role in helping chatbot to identify the entities and categorize them.In addition Figure 4.20 and Figure 4.21 demonstrates accuracy and loss graph for Bi-LSTM and Bi-GRU which gave exactly similar shape and result.

4.4.5 Performance Analysis

In Figure 4.22, we provided some examples below to prove that entities were identified correctly. We integrated named entity recognition with all other features analysed previously. We used variations in input text which in return gave us dif-

model	training		validation	
	loss	acc	loss	acc
Bi-directional LSTM	0.037	0.989	0.047	0.986
Bi-directional GRU	0.034	0.989	0.045	0.986

Table 4.9: Accuracy for training and validation data of different Deep-Learning Models for Named-Entity Recognition

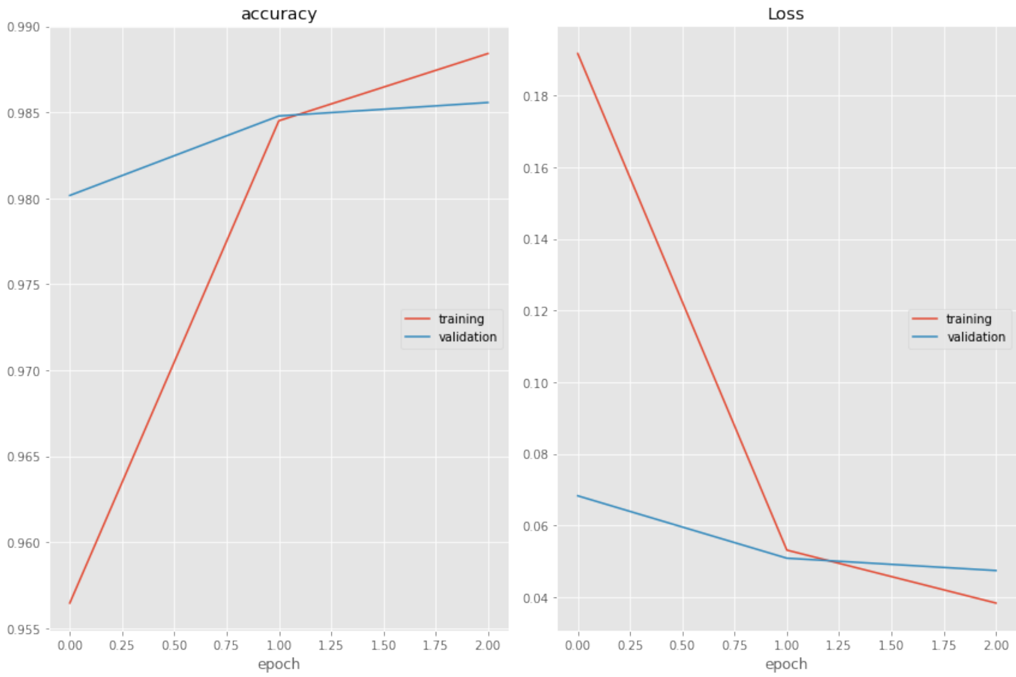


Figure 4.20: Accuracy and Loss for Named-Entity Recognition using Bi-LSTM

ferent sentiments and variety of emotion, intent and entity. We experimented with a different sentence in the first input which was nowhere related to bank issues but surprisingly it gives *"'edit_personal_details"* which is not as bad as we had expected for this particular text. In addition, all other components were accurately identified. We also experimented by giving complex sentences as input. For a chatbot to understand the sentiment, emotion, intent and entity of any complex sentence is very strenuous. Because complex sentences have one independent clause and one or more dependent clause. The meaning of one clause depends on the other. Moreover, the first clause of a complex sentence might be of positive sentiment but the second clause can be of negative sentiment. For proper utterance understanding, a conversational agent needs to identify multiple sentiments, emotions, intents and entities in a complex sentence. The second input was *"I lost my phone yesterday, so I could not help out enough to my thesis team"* which is pretty complex. But our chatbot could predict all the components of the complex sentence and predicted sentiment as negative, emotion as guilt, intent as lost_or_stolen phone and entity

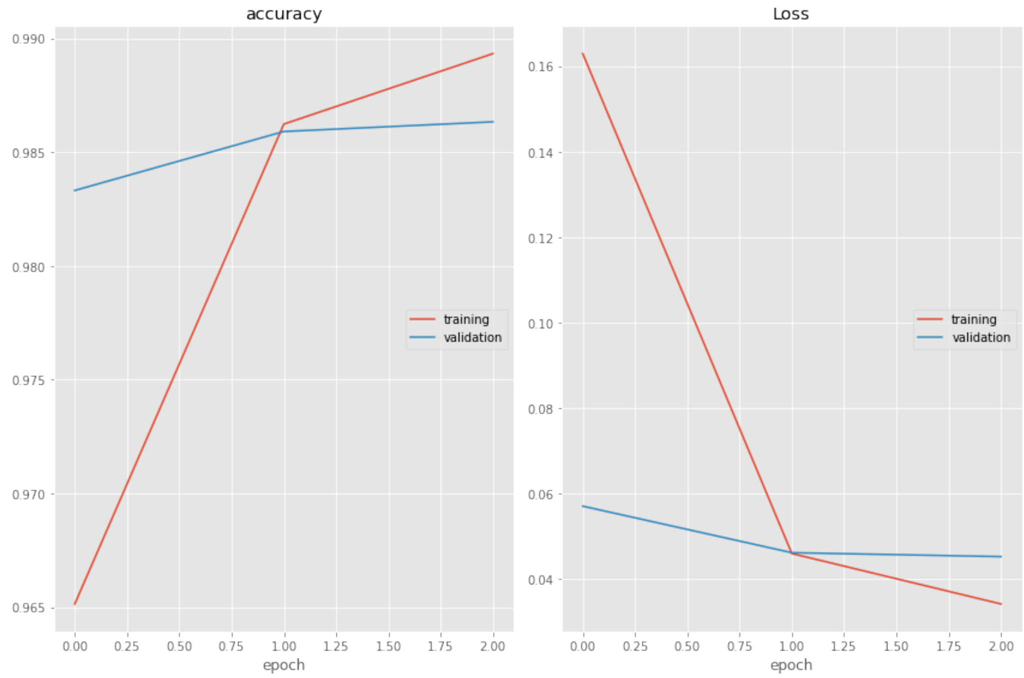


Figure 4.21: Accuracy and Loss for Named-Entity Recognition using Bi-GRU

yesterday as time. Next, we gave a bit more complex sentence which had multiple entities and it predicted all four entities. The examples also show us that our bot is now completely prepared to understand a user's given dialogue. This can be used in different sectors for understanding queries or statements of a chatbot user simply by changing the intent dataset to the desired intent domain.

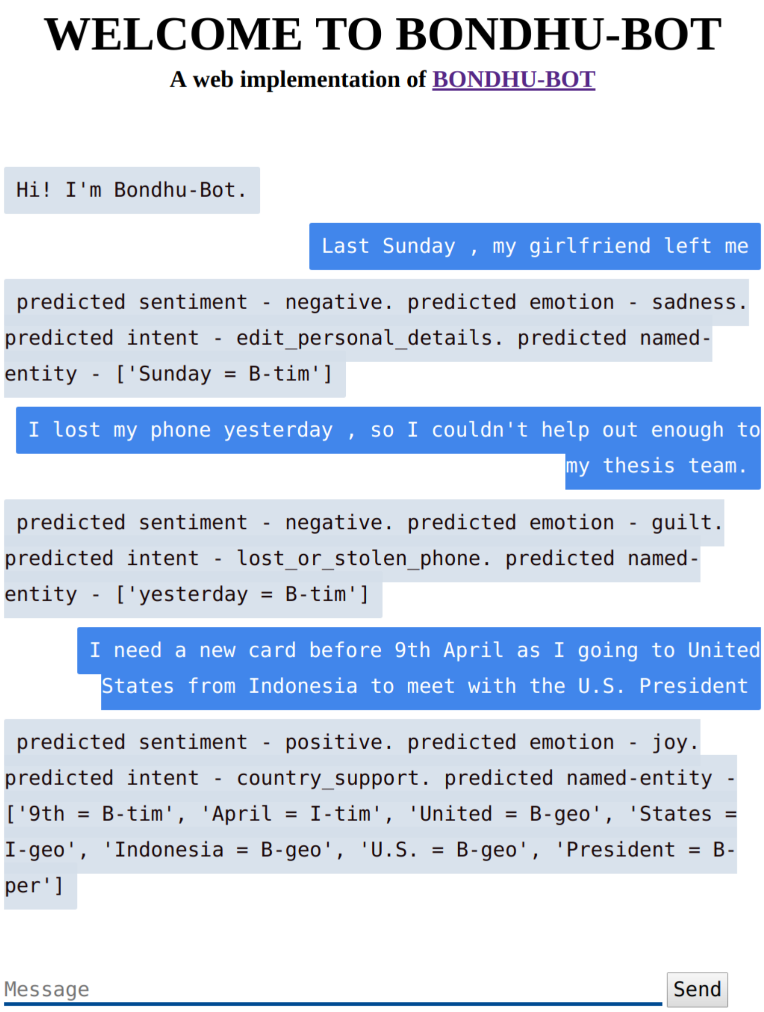


Figure 4.22: Prediction for Named-Entity Recognition

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Throughout our research, we gathered all the necessary elements for a conversational agent to recognize the *"what"*, *"where"*, *"when"* and *"how"* of an input text. This paper proposes that for a chatbot to appropriately respond to users he needs to collect all discrete information from the uttered response or dialogue of the speaker. The chatbot needs to capture the expressed sentiments, emotions as well as the intention, behind the uttered dialogue of the user in depth. We analysed all the components through application of deep learning techniques and achieved our goals in making accurate predictions. However, through our work we understood it is not at all easy to build a chatbot that can mimic a human and make conversations interactive. In this research we tried to incorporate all the functions that will help to build a human-like chatbot which has innumerous applications.

5.2 Future Work

As now our bot acknowledges personalised words and emotions, in future anyone including us can build a complete chatbot that can communicate in a humanistic manner. If the chatbot is trained with a dialogue dataset of a specific domain, it will be able to respond referring to all the analysed features we worked on in this research paper. We can vision applications of this personalised chatbot in various places from customer care centres to organisations that want to improve services through customer feedback to mental health websites or centres etc. We, specifically, are interested in developing a psychiatrist bot for mental health fighters.The pandemic has led the entire world population stuck in a corner of their room and caused severe depressions, anxiety, stress. People are unable to reach out others and express their feelings. Usually, our mental health issues are always ignored unless we reach to point of complete mental disorder. Communication helps people stay mentally healthy by reducing their piled up thoughts, opinions, anger or any suppressed emotion. It's extremely difficult to find a reliable listener and thus, we want to create an interactive bot that can help people improve their current state of mind.We would also like to work in depth on the drawbacks of keeping history of the previous conversations and able to identify the dependencies of sentences just like a psychiatrist who evaluates a person based on the conversation of different sessions. This is another reason for choosing the field of deep learning as deep learning models will help us achieve our desired goal. Furthermore, we would like to move towards unsupervised learning where the chatbot can dynamically understand and generate responses without training. The entire work is a lot challenging and will require proper setup for training and large quantities of datasets.

Bibliography

[1] "Why Deep Learning over Traditional Machine Learning?". [Online]. Available: https://towardsdatascience.com [Accessed: 4-Jan-2021].

[2] "How to Build a Chatbot That Can Interact Like Humans." [Online]. Available: https://shanebarker.com/blog/build-a-chatbot. [Accessed: 4-Jan-2021].

[3] "The Virus Changed the Way We Internet." [Online]. Available: https://www.nytimes.com/interactive/2020/04/07/technology/coronavirus-internet-use.html. [Accessed: 4-Jan-2021].

[4] "Top 10 Reasons: Why Your Business Need A Chatbot Development."[Online]. Available: https://chatbotsjournal.com [Accessed: 4-Jan-2021].

[5] "Deep Learning Chatbots: Everything You Need to Know." [Online]. Available: https://hackernoon.com. [Accessed: 4-Jan-2021].

[6] "The Future of Psychiatry: Telehealth, Chatbots, and Artificial Intelligence." [Online]. Available: https://medicalfuturist.com/future-of-psychiatry/. [Accessed: 4-Jan-2021].

[7] "Advances in Conversational AI. (n.d.)." [Online]. Available: https://ai.facebook.com/blog/advances-in-conversational-ai/. [Accessed: 4-Jan-2021].

[8] "Machine Learning Vs Deep Learning: It's time to know the difference" [Online]. Available: https://www.mygreatlearning.com/blog/is-deep-learning-better-than-machine-learning/. [Accessed: 4-Jan-2021].

[9] "Google is Working on Chatbots Which Can Engage in a More Genuine, Human Conversation." [Online]. Available: https://www.socialmediatoday.com/news/. [Accessed: 4-Jan-2021].

[10] B. A. Shawar, and E. Atwell, "Chatbots: Are they Really Useful?" LDV Forum. 22. 29-49.

[11] J. Jia, "CSIEC (computer simulator in educational communication): An intelligent web-based teaching system for foreign language learning." In Kommers, P. and Richards, G., ed-itors, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunica-tions 2004, pages 4147–4152, Chesapeake, VA. AACE press.

[12] J. Jia, "The study of the application of a web-based chatbot system on the teaching of foreign languages." In Proceedings of the SITE2004 (The 15th annual conference of the Society for Information Technology and Teacher Education), pages 1201–1207. AACE press.

[13] D. Kawade, and K. Oza,"Sentiment Analysis: Machine Learning Approach." International Journal of Engineering and Technology. 9. 2183-2186. 10.21817/ijet/2017/v9i3/1709030151.

[14] M. Y. Hossain, I. Hossain, M. Banik, M. I. Hossain, and A. Chakrabarty, "Embedded System based Bangla Intelligent Social Virtual Robot with Sentiment Analysis." 10.1109/ICIEV.2018.8641023.

[15] D. Lee, K.-J. Oh and H.-J. Choi, "The chatbot feels you - a counseling service using emotional response generation," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, 2017, pp. 437-440, doi: 10.1109/BIGCOMP.2017.7881752.

[16] M. Su, C. Wu, K. Huang, Q. Hong and H. Wang, "A chatbot using LSTM-based multi-layer embedding for elderly care," 2017 International Conference on Orange Technologies (ICOT), Singapore, 2017, pp. 70-74, doi: 10.1109/ICOT.2017.8336091.

[17] S. Hochreiter, J. Schmidhuber, "Long Short-term Memory." Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[18] A. Karpathy, J. Johnson, L. Fei-Fei, "Visualizing and Understanding Recurrent Networks." Cornell Univ. Lab.

[19] V. A. Bhagwat, "Deep Learning for Chatbots". Master's Projects. 630.

[20] F. Acheampong, C. Wenyu, and H. Nunoo-Mensah, "Text-Based Emotion Detection: Advances, Challenges and Opportunities." Engineering Reports, 2020.

[21] M. Balaji, N. Yuvaraj, "Intelligent Chatbot Model to Enhance the Emotion Detection in social media using Bi-directional Recurrent Neural Network." Journal of Physics: Conference Series. 1362. 012039. 10.1088/1742-6596/1362/1/012039.

[22] V. Yadav, S. Bethard, "A Survey on Recent Advances in Named Entity Recognition from Deep Learning models." Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, August, 2018, pages 2145-2158. Association for Computational Linguistics.

[23] L. Meng, M. Huang, "Dialogue Intent Classification with Long Short-Term Memory Networks." 10.1007/978-3-319-73618-1_4.

[24] M. Awad, R. Khanna, "Support Vector Machines for Classification." 10.1007/978-1-4302-5990-9_3.

[25] T. Mullen, N. Collier, “Sentiment Analysis using Support Vector Machines with Diverse Information Sources.” 412-418.

[26] R. Deshmukh, D. K. Kirange, “Emotion Classification of News Headlines Using SVM.”

[27] V. A. Kharde, S. Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques.” International Journal of Computer Applications 139(11): 5-15, April 2016.

[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis.” The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

[29] ‘Twitter_sentiment”. [Online]. Available: `https://www.kaggle.com/ywang311/twitter-sentiment`. [Accessed: 4-Jan-2021].

[30] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation.” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[31] “Emotion Detection and Recognition from Text Using Deep Learning.” [Online]. Available: `https://devblogs.microsoft.com/cse/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/`. [Accessed: 4-Jan-2021].

[32] V. V. Ramalingam, A. Pandian, A. Jaiswal, and N. Bhatia, “Emotion detection from text.” Journal of Physics: Conference Series. 1000. 012027. 10.1088/1742-6596/1000/1/012027.

[33] O. Bruna, H. Avetisyan, and J. Holub, “Emotion models for textual emotion classification.” Journal of Physics: Conference Series. 772. 012063. 10.1088/1742-6596/772/1/012063.

[34] P. Ekman, “Basic Emotions.” 10.1002/0470013494.ch3.

[35] A. Syaekhoni, S. Seo, Y. Kwon, “Development of Deep Learning Models for Multi-class Sentiment Analysis.” Journal of Korea Information Technology service. 16. 149. 10.9716/KITS.2017.16.4.149.

[36] K. R. Scherer, H. G. Wallbott, “Evidence for universality and cultural variation of differential emotion response patterning.” J Pers Soc Psychol. 1994 Feb;66(2):310-28. doi: 10.1037//0022-3514.66.2.310. Erratum in: J Pers Soc Psychol 1994 Jul;67(1):55. PMID: 8195988.

[37] J. Liu, Y. Li, M. Lin, “Review of Intent Detection Methods in the Human-Machine Dialogue System”. Journal of Physics: Conference Series. 1267. 012059. 10.1088/1742-6596/1267/1/012059.

[38] I. Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić, “Efficient Intent Detection with Dual Sentence Encoders”.

[39] W. Khan, A. Daud, F. Alotaibi, N. Aljohani, S. Arafat, "Deep recurrent neural networks with word embeddings for Urdu named entity recognition". ETRI Journal. 42. 10.4218/etrij.2018-0553.

[40] "Annotated Corpus for Named Entity Recognition," [Online]. Available: https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus. [Accessed: 4-Jan-2021].

[41] "A Gentle Introduction To The Rectified Linear Unit (Relu)." [Online] Available: `https://machinelearningmastery.com/`. [Accessed 3 January 2021].

[42] "A Gentle Introduction To Dropout For Regularizing Deep Neural Networks." [Online]. Available: https://machinelearningmastery.com/. [Accessed 3 January 2021].

[43] "Light On Math ML: Intuitive Guide To Understanding Glove Embeddings." [Online]. Available: `https://towardsdatascience.com/`. [Accessed 3 January 2021].

[44] "Sigmoid Function." [Online] Available: `https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function[Accessed3January2021].`

[45] "What Is Stemming?" [Online]. Available: `https://searchenterpriseai.techtarget.com/definition/stemming`. [Accessed 3 January 2021].