

Contextual Healthcare Chatbot using Deep Neural Network

MSc Research Project
Masters in Data Analytics

Suryakanta Sahu
Student ID: X20141513

School of Computing
National College of Ireland

Supervisor: Dr. Majid Latifi

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Suryakanta Sahu
Student ID:	X20141513
Programme:	Masters in Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Majid Latifi
Submission Due Date:	31/01/2022
Project Title:	Contextual Healthcare Chatbot using Deep Neural Network
Word Count:	7144
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Suryakanta Sahu
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contextual Healthcare Chatbot using Deep Learning

Suryakanta Sahu
X20141513

Abstract

As per the popular saying, a healthy society is a wealthy society, hence healthcare is and always will be among the most vital components of any civilization. With our planet's population growing at a 1.2 percent yearly rate, the world's healthcare system is under tremendous pressure than ever. Also, the changing lifestyle, and multifold increase in pollution are adding fuel to this existential crisis. In this strenuous scenario, technology can offer a helping hand in many ways, and one of them is virtual assistant or chatbot systems. Chatbots are a blend of Artificial Intelligence, Natural Language Processing and big Data to provide accurate information on fingertip. As a major percentage of the world's population lives away from a regular medical facility, chatbots can be acted as a lifesaver in many situations by easily accessing them through smartphones. People are also stimulated to use a chatbot for intimate queries rather than visiting a doctor in person because of the anonymity and privacy conferred by chatbots. In this study, a contextual chatbot empowered by deep learning algorithm and natural language processing techniques was developed and the model's performance was evaluated against a Naive Bayes model and numerous state-of-the-art solutions. For training the conversational system, data was gathered from a number of open-source websites like Wikipedia, WebMD etc. This developed solution also has context retention capabilities, which means it retains certain information from a prior query so that the model can respond to subsequent inquiries more efficiently and with less input from the end user. The suggested model is anticipated to assist users with healthcare related enquiries instantly and without the need for an active internet connection. Upon evaluating the model with a number of accuracy metrics, a train accuracy of 98% and validation accuracy of 81% was achieved.

Keywords – Healthcare, Deep Neural Network, Naive Bayes, Chatbot, Conversational System

1 Introduction

Productivity and time management have assumed primacy in modern human's day-to-day routines as a consequence of constrained time schedules and improved socioeconomic conditions. People prefer instant answers available at their fingertips over the conventional professional consulting procedure, which involves a great deal of time and effort. Numerous polls conducted recently have revealed the same conclusion: (Vailshery; 2021; Dutt et al.; 2020; Gupta et al.; 2021). In the domain of medical and healthcare, the tendency is even more indulgent towards digital solutions since it saves time, cost, and assures anonymity. Because visiting a doctor offline necessitates a previous appointment and is highly costly. As a result, individuals are turning to the internet for answers to minor

problems. Apart from the cost and time factor, this results in many other advantages such as, reduction of communication gap between doctor and patient which might occur in a face to face meet up and the most important one i.e., privacy and discreteness. In general, most individuals are uneasy revealing their personal worries with others, but they have no such qualms typing in online owing to the machine's unbiased nature. Though the online solutions have many benefits, one of the biggest drawbacks of it is that, they require the clients to have an operational internet connection in order to obtain answers, which is inconvenient in many instances. Under such challenging situations, the conversational systems/chatbots, empowered with their 24×7 offline accessibility and fast reply, might be a saviour to many users. Apart from that, the users might have access to chatbots at all times by installing them on their phone, tablet, or computer. Furthermore, chatbots might be trained in a variety of medical disciplines to assist users with a variety of questions; otherwise, patients would have to see many doctors who specialize in different subjects.

Although there are several virtual assistant platforms already being used in the healthcare area, most of them are generally employed for customer service operations such as scheduling appointments or enabling payment systems (Nursetyo et al.; 2018; V et al.; 2021). The goal of this study is to create a contextual conversational system/chatbot that uses a Deep Neural Network (DNN) and Natural Language Processing (NLP) techniques to understand the context of the user discussion and only respond with answers that are relevant to the present context. Therefore, it allows the solution to remember the user's prior questions, ensuring that the following response is appropriate. Another significant attribute of this proposed solution is that, the knowledge base of this conversational system is in JSON format, which can be easily maintainable and highly scalable. In addition, the solution's core is built on a multilayered deep neural network with appropriate neurons and activation/loss functions that is easily adaptable as requirements evolve. In addition to the deep neural network model, a Naive Bayes model was also developed to compare the efficiency and performance. Despite the fact that many ongoing and prior studies on conversational systems have been undertaken, the fundamental difficulty with most of them is their accuracy and capacity to preserve the context of the discussion. For example, though the healthcare chatbot created by Astuti et al. (2021) had a decent accuracy in terms of answering the user query, it was unable to remember the context of the progressing conversation. This proposed solution aims to overcome the inadequacies of the existing conversational systems with regard to their accuracy and ability to retain the discussion context.

1.1 Research Question

The research question of this project as following :

How accurately an artificial intelligent chatbot solution powered by deep learning can answer health-related user questions and be able to recall the context of the ongoing conversation?

1.2 Research Objectives

The following research objectives are expected to be achieved by conducting this project:

1. Study and detailed review of previous state-of-art research papers related to conversational systems and question answering and Chatbot.

2. Proposing a conversational system using deep learning techniques.
3. Implementation of the proposed system using Python language and various packages of Keras and NLTK.
4. Evaluation of the developed conversational system in terms of accuracy and responsiveness.

The primary contribution of this study towards the existing researches on conventional system is the novel approach of applying a multi-layered deep neural network and the ability to retain the context of the ongoing conversation.

The remaining sections of this report organized as follows: The Section 2 delves into the details of a number of previous and continuing studies on conversational systems, while the Section 3 explains the proposed methodology. The techniques and frameworks used in the project are briefly described in Section 4. The execution of the suggested method is discussed in depth in Section 5. The Section 6 outlines the evaluation methods and results of the implemented system. Finally, Section 7 delineates the future work and concludes the report.

2 Related Work

Due to the advancement of artificial intelligence technology, particularly deep learning techniques, a lot of ongoing researches are being conducted in the field of conversational system. A comprehensive literature survey was undertaken to gather knowledge of modern trends and aspects in the Chatbot domain, and the most successful experiments are discussed here. The related work section is primarily segregated into three sub-categories, i.e., Chatbots using Deep Learning Models as Core, Chatbots using Transferred Learning and advanced Deep Learning Models as Core, and Conversational Systems with Contextualization Capability.

2.1 Chatbots using Deep Learning Models as Core

To make the interaction between the AI machine and users more humane, Carranza et al. (2019) have developed a chatbot which takes user input message as well as the live video of the user's face to provide an appropriate response. Authors have used a multi layered deep neural network using tensorflow for training the text messages and images from the live video. The tensorflow based facial expression classifier was able to categorize the user's face into three categories i.e., happy, neutral and hurt. A total, 271 images were trained and accuracy of 67%, 17%, 57% were yielded for happy, neutral and hurt facial expressions respectively. Though the solution was able to respond emotionally intelligent responses, but the accuracy of the chatbot was not so great and the training dataset size also very limited. The efficiency and accuracy of the solution could be significantly increased by taking a bigger dataset and applying more complex NLP techniques.

In another interesting research, Prasetyo and Santoso (2021) have utilized intents as the training data to create a conversational system. Intents dataset contained a set of conversations/sentences which are categorized into different tags. For example, sentences like 'Hi', 'hello' and 'how are you' etc could be categorized as greetings, similarly, all the sentences in the dataset were classified into different tags. The model incorporated a Recurrent Neural Network (RNN) structure to train the input data. The dataset used had

around 800k corpus and cleaned and transformed appropriately by using NLP techniques before being fed into the RNN model. For evaluating the model, the authors had utilized K-fold cross validation with $k=5$. Upon evaluation, the RNN model achieved a maximum accuracy of 81%. To make the model more generic and effective, the solution might be enhanced by include more diverse intents to the input dataset.

Raundale and Sawale (2021) have created a closed domain chatbot using Sequential neural networks. In addition to the neural network, the researchers applied the Stochastic gradient, ReLU, and Softmax activation functions for classification. The training data was collected through a survey using google forms and pre-processed using various NLP and regular expression packages and techniques. For data pre-processing the authors have used NLP techniques like tokenization, stemming and bag of Words (BoW). The three layered model was executed with 200 epochs and achieved an exceptional 98.5% of training accuracy. The model was then deployed to a Flask application to integrate with an interactive user interface. Though the solution achieved an outstanding training accuracy, the input dataset is quite small, hence the model would struggle with fresh and previously untrained data.

The Covid-19 pandemic has led to prolonged quarantine world-wide, which has a dreadful effect on human psychology and mental health. Sriram et al. (2021) in this study, have developed a chatbot system which asks the users under quarantine various questions at predetermined intervals and plays music to match their emotion. If the user does not respond to the circumstance, the webcam is triggered instantly, and emotional analysis using sentiment analysis interprets their face motions, encouraging them to de-stress and spread the message to other individuals. The proposed solution employed a pretrained MobileNetV2 model with customized Dense and Flatten layers. The model had achieved validation accuracy of 90% was able to detect the user's mood through the live video feed and provided the desired responses. The researchers have deployed the system with Android OS and in future aimed to integrate it with IoT sensors for real-time usage.

2.2 Chatbots using Transferred Learning and Advanced Deep Learning Models as Core

Deepika et al. (2020) in this study have created a conversational system namely; joliiity chatbot to cheer-up the users by providing motivational suggestions during tough times. For training purpose, an intent based dataset with more than 100 intents was used. To train the input dataset, an open-source transferred learning RASA model was employed. For implementing the model, a five layered modular architecture was designed, i.e., Data module, Domain module, Stories module, Credential module, and Deployment module. For evaluating the module, accuracy, and confusion matrix metrics were used. The experimental results showed a validation accuracy of 90%. The authors have deployed the solution into Telegram to verify the model with real-time user queries. As the dataset utilized in the solution is very small and domain limited, the researchers have aimed to use a larger dataset fetched from Quora to build an even more efficient conversational system.

In another interesting research, Yin et al. (2019) have proposed an innovative conversational system named Evebot, which uses sequence to sequence mechanism for the response generation. The solution is made up of three different modules, each with its own set of objectives. The deep learning based Bi-LSTM was trained with a psycholo-

gical corpus dataset and utilized for identifying the negative emotions in the users. The model was trained with an input word vector containing around 100,000+ corpus. Upon training the data with a 90:10 Train-Validation split, and with appropriate number of epochs, the researchers were able to obtain a F1 Score of 92.8 percent and Precision score of 90.91 percent. Furthermore, the authors have compared the performance of the Evebot chatbot with other market leading conversational systems. Although, the chatbot had attended excellent accuracy, but it was incapable to memorize details from the previous user discussions.

Lately, implementations of chatbots have been widely regarded across numerous domains and industries. Quan et al. (2019) have developed a conversational system to facilitate meaningful and efficient interaction with potential customers in the real estate application. The solution incorporated a multi-task DNN technique to understand the user queries and to respond with proper answer or reverting questions. The researchers mainly have used Bi-GRU structure enhanced by a dense connection method for feature extraction purpose. To create embedded information at the character level, the authors also integrate word embedding technique with a Char-CNN layer. For the input training data, the researchers have used a dataset with more than ten thousand corpus which was accumulated from various real estate advertisements. For evaluating the model, the authors have used F1 score, precision, and recall. The proposed solution had obtained 86% F1 score, 85% precision and 85.8% recall score.

In real-world conversation, people are more likely to have use their own distinct manner of speaking style, which makes the conversation more human. Chatbot models educated on plain and pre-processed text are likely to miss these intricacies. By utilizing convolutional transformer, Khadilkar et al. (2021) offered a way to embody these peculiarities in the form of a conversational system using a new Natural Language Generation framework. The researchers have used WhatsApp conversation text containing emojis, foreign languages, and customized messaging styles as input dataset to train and develop this solution. The employed convolutional transformer analyses the last 9 lines to predict the next dialogue and uses two fully connected layers, two encoder decoder modules. For evaluation, authors have used accuracy and IOR metrics and observed train and test accuracy as 87% and 72% respectively. But more crucially, the model was able to respond with emojis and personalized phrases. The solution only worked fairly with small sentences, and with the increasing number of words the conversation became more incoherent. The authors suspect, by employing BEAM search instead of the currently used greed search, this issue could be solved.

Makhkamova et al. (2020) in this study, have developed a multi-chatbot to overcome the shortcomings of usage of a single chatbot. A single chatbot has the limitations in terms of response time and the variety of information it can be trained upon, whereas a multi-chatbot can have large-scale coverage and faster response time. To create a multi-chatbot broker system the researchers have proposed a solution with four major modules namely; User, Video tutor assistant, broker, and chatbots. The chatbots are first trained with data from various domains and then the broker uses deep learning for classifying the user's query and send it to the most appropriate chatbot which is expert in that particular domain. The proposed system used Word2Vec for feature extraction and skip-gram was used for word embedding. Researchers have experimented the model using Naive Bayes, SGD, Bi-LSTM, CNN etc. After evaluation, the Naive Bayes found to be most accurate and effective, with around 90% train accuracy and 80% test accuracy. The solution could be further improved by implementing customized recommendation capability based on

the end-user’s query history.

Nowadays, a lot of virtual assistant systems have been developed to automate conversation between users and machines, but most of them are available with English language. Languages with very limited digital footprint and popularity have very limited number of virtual assistant support. In this interesting research, Dasanayaka and Warnajith (2020) have developed a conversational solution in Sinhala language using deep learning. Because the input data was not in English, a feature extraction approach based on regular expressions was utilized. Long short term memory (LSTM) and RNN were incorporated to create an Intent Mapping model. The model yielded an impressive 90% of validation accuracy. The model performance could be further improved by utilizing Word Vector or Bag of Words (BoW) algorithms on the input sentences for creating the sparse matrix.

2.3 Conversational Systems with Contextualization Capability

To add context to the conversations, Keerthana et al. (2021) have developed a conversational system using Deep Reinforcement Learning technique. The proposed system employed Neural Machine Translation (NMT) and Bidirectional Recurrent Neural Network (BiRNN) for processing and training lengthier sentences. For implementing context, the researchers have utilized deep reinforcement learning models like Deep Q Neural Network and Distributional Reinforcement Learning with Quantile Regression. A big dataset with 8000+ conversations was used as the training data and afterwards, accuracy, performance of each model were analysed and compared. Upon comparing the results, the model employing DR-DQN was found to be most accurate and efficient, with 88% of validation accuracy. The model was also tried with different other dataset and was observed to be equally efficient on them.

With the emergence of encoder-decoder and Deep Neural Network, the conversational systems have become more intuitive and precise. DNN with LSTM sequence-to-sequence (SEQ2SEQ) have lately demonstrated remarkable response generation capability. In general, the seq2seq model uses the maximum likelihood estimation (MLE) principle to generate an appropriate response. MLE, on the other hand, finds it challenging to approach the chatbot’s development goal: training a computer to communicate with humans and returns hardcoded responses like “I didn’t get that” while responding to unknown queries. But, while dealing with very long conversations, the DNN chatbots often lose the context of the topic and prone to provide irrelevant and hard-coded default messages. To overcome this issue, Tran and Le (2021) have developed a chatbot solution using reinforcement learning along with a policy gradient rewarding mechanism. For training data, the authors have used DailyDialog dataset contains around 13,118 dialogues. The researchers have used BELU score to evaluate the coherence and accuracy of the models. Upon comparing, the proposed solution was observed to be more coherent with 43 percent average BELU score, whereas the baseline DNN model had 35 percent BELU score. To improve the proposed system, the authors have aimed to explore other state-of-the-art reinforcement learning models as well as more effective reward mechanism.

Kandpal et al. (2020) have created a virtual chatbot solution which was trained on healthcare related dataset to help users with health-related queries. In this research, the authors have leveraged NLP techniques and neural networks to train the input JSON data. After tokenizing the training data, Lancaster Stemmer package was utilized for data pre-processing. The model’s predictions were found to be highly optimum and contextual after testing it with real-time user inquiries. The model’s performance and

domain could be further enhanced by utilizing a more deep neural network and making the input data to undergo with various other stemming techniques like Porter Stemmer and Lemmatization. The training dataset’s size might also be extended and improved so that the conversational system can respond to a wider range of user questions.

The below table 1 summarizes the methods used, evaluation parameters and problems addressed in various state-of-art researches.

Table 1: Summary of the State-of-the-art studies related to Conversation systems

Author(s)	Methods Used	Evaluation Parameters	Problems Addressed/Advantages
Makhkamova et al. (2020)	Bi-LSTM, CNN, Naive Bayes, SGD, Skip Gram	Train and Test Accuracy, Loss	Improve response time and accuracy by applying multiple chatbots
Khadilkar et al. (2021)	Convolutional Transformer, Encoder-Decoder	IOR, Accuracy, Loss	The solution was able to comprehend and respond to emojis and customized foreign language texts.
Tran and Le (2021)	Neural Reinforcement Learning, DNN, Policy gradient methods to reward sequences	Coherence, BLEU	In long conversations, the chatbot was able to retain the context and avoided hard-coded replies to unknown answers.
Keerthana et al. (2021)	Natural Machine Transmission (NMT), Bidirectional Recurrent Neural Network (BiRNN), Deep Reinforcement Learning, Quantile Regression	BLEU score, Cumulative reward plot, Cumulative Regret Plot	The DRL algorithm enables the chatbot to distinguish between right and wrong responses by implementing a cumulative reward system.
Dasanayaka and Warnajith (2020)	Intent Mapping Model, LSTM, RNN	Precision, Recall, F1 Score, Confusion Matrix	The solution was able to identify and understand non-English language like Sinhala and response in the native language.
Sriram et al. (2021)	MobileNetV2, Customized Dense layers	Train and Test Accuracy, Loss	The pretrained weightage of the MobileNetV2 model enabled to increase the model accuracy substantially.
Tran and Le (2021)	Reinforcement Learning, Policy Gradient Reward Mechanism	BELU Score	The model was able to perform really well while dealing with long conversations and avoided giving hard coded responses.

3 Proposed Research Methodology

This section explains the approach that was utilized to address the study’s primary findings, architectural technological design, and data analysis procedure. The proposed

conversational system is a retrieval-based conversational solution which works on the principle of being trained from a predefined question-answer pairs (knowledge base) and taught to offer the best possible response from the preset replies. To find the most relevant answers, this retrieval-based chatbot system employs techniques such as keyword matching, and deep learning. The suggested method was based on employing appropriate Natural Language Processing techniques to pre-process the input dataset, and then create a word vector to enable it to be utilized as input to a dense deep neural network model. Thereafter, to train the pre-processed data, a sequential multilayered deep neural network with appropriate number of neurons and activation function was developed. The trained model was then evaluated and further improved for best accuracy and minimum loss using a variety of optimization strategies. At the end, the developed model was enabled to accept raw user queries and answer appropriately. It employs the approach depicted in Figure 1.

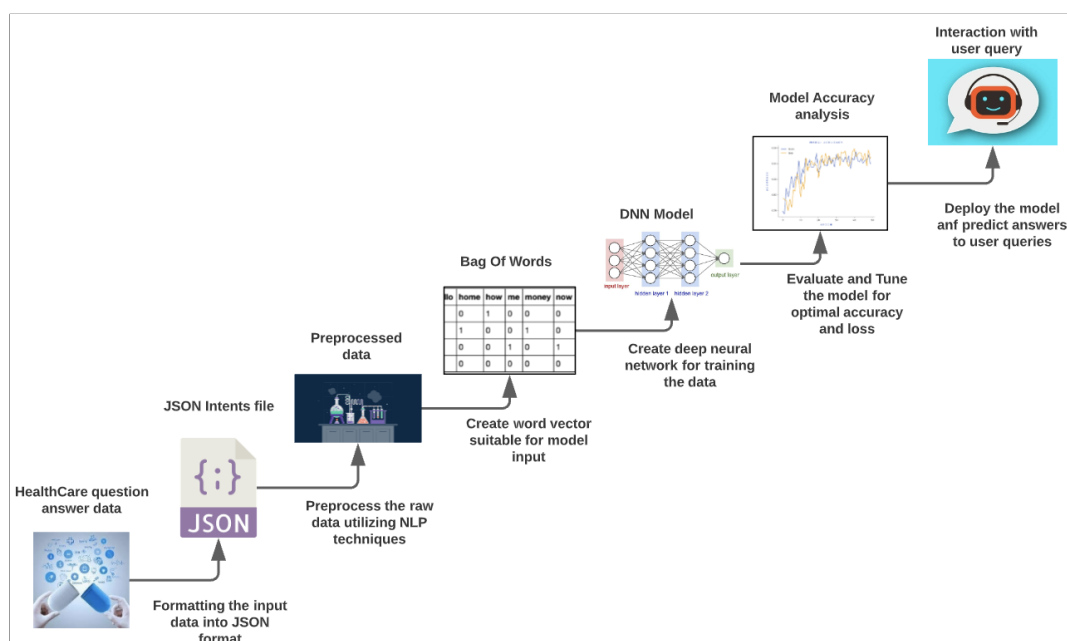


Figure 1: Methodology Approach

3.1 Data Acquisition and Initialization

As there was no conventional, standard healthcare related dataset readily available, the information to create a knowledge base was collected manually. The data utilized in this study was gathered in the form of question-answer pairs from a variety of sources, including Wikipedia, WebMD, and others. After data acquisition, the obtained data was stored in JSON format under the name intents.json. The knowledge base JSON file contains around 200+ question answer pairs and has four sub-tags namely “tag”, “patterns”, “responses” and “context_set”/“context_filter”. These sub-tags holds the below details and exhibited in Figure 2.

tag - It is the primary tag to which the question(pattern) and answers(responses) pairs belong.

patterns - It contains the questions (keywords) which are matched to the user queries.

responses - It contains the predefined answers to the corresponding patterns and used

as response to the user queries.

context_set/context_filter - It holds the keyword for context of the user query.

```
{
  "tag": "Fever",
  "patterns": ["How do you treat a mild Fever?", "what to do if i get a mild fever?",
    "", "fever"],
  "responses": ["To treat a fever at home: 1)Drink plenty of fluids to stay hydrated.
    blanket if you feel chilled, until the chills end. 4)Take acetaminophen (Tylenol, o
    5) Get medical help if the fever lasts more than five days in a row."],
  "context_set": "fever"
},
{
  "tag": "Symptoms",
  "patterns": ["what are it's symptoms", "symptoms of cold", "symptoms"],
  "responses": ["Symptoms of fever are : 1)Temperature rise 2)Headache 3)Body pain"],
  "context_filter": "fever"
},
{
  "tag": "Nasal Congestion",
  "patterns": ["How do you treat nasal Congestion?", "what to do if i get a nasal con
    nasal congestion?", "what to do if i have a blocked nose?", "How do you treat a blo
    last?"],
  "responses": ["When you're stuffed up, focus on keeping your nasal passages and sin
    you can: 1)Use a humidifier or vaporizer. 2)Drink lots of fluids. This will thin ou
    sinuses. 3)Place a warm, wet towel on your face. It may relieve discomfort and open
    "context_set": "Nasal Congestion"
},
}
```

Figure 2: Structure of the intents.json

3.2 Data Preprocessing and Transformation

The raw data was first cleaned to remove any undesired characters in order to reduce noise. Unwanted characters like punctuations and English stop words were removed using regular expression and NLP techniques. For the purpose of simplicity and to facilitate smooth processing, the cleaned data was then converted to lowercase. Afterwards, tokenization was performed on the raw text to break down them into chunks called tokens. By evaluating the sequence of words, tokenization aids in deciphering the meaning of the text. The tokens were then stemmed to get them back to their root form, which reduces the text dimensionality by standardizing it. As the machine learning models don't understand raw natural language text, the pre-processed text were converted into vectors of numbers by using Bag of Word (BoW) feature extraction technique. The user's input queries were also subjected to similar data pretreatment procedures in order for the model to handle them efficiently.

3.3 Deep Neural Network Model Designing

A multilayered deep neural network (DNN) was developed and implemented to train the input data. A deep neural network comprises more than three layers, in contrast to a basic neural network, which has lesser number of layers. The ability to learn and perform the classification improves as the number of layers increases. It's more sophisticated than a simple neural network, as it can process input data faster and make better forecasts given the data it's been taught on. In a DNN, the input layer is not linked to the output layer directly; instead, it is connected to the intermediate layers, and each layer is connected to the subsequent layers. The DNN incorporated in this model has 4 intermediate fully-connected layers, as well as one input and one output layer. The input layer has the shape same as the number of columns in the input document matrix, whereas the intermediate dense layers have 10 neurons each. The output layer employs softmax activation function, and RELU activation functions were used in the intermediate layers.

3.4 Evaluation

The performance and efficiency of the model was calculated by using various accuracy metrics like Train/Test accuracy and loss. The TensorFlow framework reserves part of the input data as validation data and verifies the model's accuracy as it's being trained. After the model parameters are defined, the accuracy score is calculated as a percentage value, and it represents how close the model's forecast is to the actual data. The cross-entropy loss function was used to optimize the model by calculating the sum of errors in both training and test data. The loss value indicates how fairly the model algorithm performs after every optimization iteration. The accuracy and cross-entropy loss can be defined as below:

$$Accuracy = \frac{\text{No of correct predictions}}{\text{Total no of predictions}} \quad \text{Cross-entropy} = - \sum_{i=1}^n \sum_{j=1}^m y_{i,j} \log(p_{i,j})$$

The accuracy and loss value can be interpreted as below:

- Great Accuracy & low Loss (Ideal Case) - Indicates that the model has made small errors on a few data.
- Great Accuracy & great loss - Indicates that the model has made small errors on a lot of data.
- Low Accuracy & low Loss - Indicates that the model has made big errors on a few data.
- Low Accuracy & great Loss - Indicates that the model has made big errors on a lot of data.

In addition to the accuracy and loss, evaluation measures like F1 score, Precision, and Recall were used to assess the efficiency of the developed models. Precision is the percentage of relevant data points found among the recovered instances, whereas recall/sensitivity is the percentage of relevant instances found. In the most basic form, Precision is the ratio of True positives to all positives, i.e., the percentage of the affirmative identifications that were correct and Recall is the percentage of true positives that were successfully ascertained. The average of precision and recall is calculated in F1 score in order to create a single statistical value. The F1 score is particularly effective in the situation of imbalanced data. As the models, in this study deals with multiclass classification problem, the weighted average of the precision, recall, and F1 score was considered. The mathematical formula for Precision, Recall, and F1 score are provided below.

$$Precision = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Positive}(FP)} \quad \text{Recall} = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Negative}(FN)}$$

$$F1 \text{ Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

4 Design Specification

Figure 3 shows the design specification diagram that describes the processes and architecture in order to develop the proposed model. The whole design and implementation is executed on Jupyter notebook and Google Colab in order to harness the built-in GPU provided by Google. For data pre-processing and transformation libraries like NLTK, Regex are used. Following that, a deep neural network was created, and the pre-processed data was trained upon it. For developing the DNN, Python was chosen as the core programming language because of its ease of use and the abundance of assistance libraries available, such as TensorFlow, Keras, JSON, NLTK, Numpy, and Pandas etc. In addition, Keras Sequential wrapper is utilized which is an abstraction framework build on top of TensorFlow to make developments easier and faster, while staying entirely compatible with it. After the model is trained, appropriate hyper-tuning techniques are implemented to increase the accuracy and minimize the loss. The efficiency of the trained model is visualized on the tensorboard with various plots like Accuracy vs Iteration and Loss vs Iteration etc. Finally, the model is integrated with the conversational system’s UI to interact with user queries.

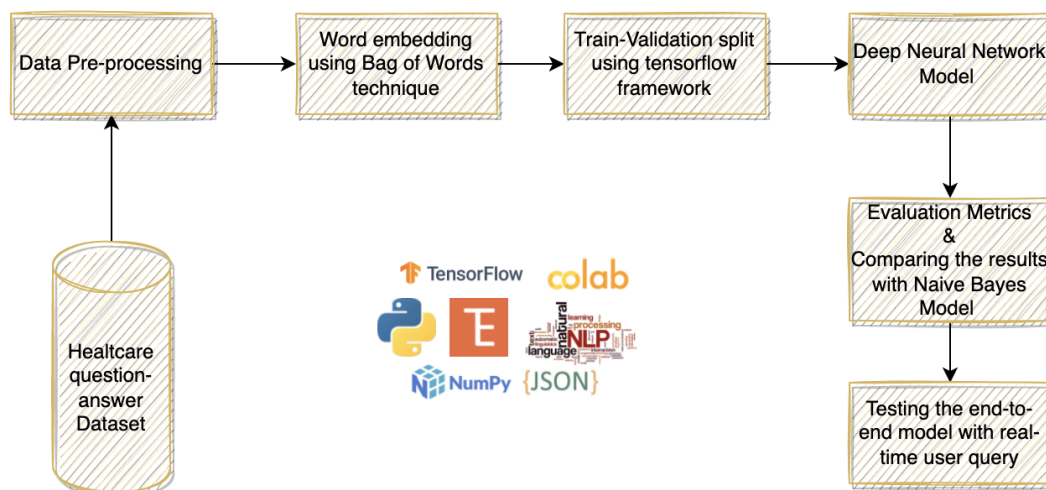


Figure 3: Design Specification

5 Implementation

This section provides a detailed depiction of the implementation of the proposed approach/methodology.

5.1 Setup

The data loading, pre-processing, feature extraction, and evaluation part of the project were performed on Jupyter Notebook which is part of the Anaconda Distribution. The host machine was a MacBook Pro laptop with a configuration of 3.5 GHz Dual-Core Intel Core i7 processor and 16 GB of RAM.

For the development and training of the core neural network model, Google Collaborator was utilized due to the availability of Graphics Processing Unit (GPU). In general,

GPUs are more efficient to train the multi-layered deep learning networks because of their ability of parallel processing and intense computational efficiency. All the dependency libraries like Numpy, Pandas, tensorflow, keras, re, random, NLTK, and json were installed in the working environment and then imported to the project space. Python was used as the programming language to write and execute high level code.

5.2 Data Handling

At first, healthcare related question answer pairs were collected from various websites like, WebMD, Wikipedia etc and then stored in JSON format. The JSON file (intent.json) had four tags namely; tag, patterns, responses, context_set/context_filter. The tag contains the classes (for e.g., Fever, Cold, Headache etc), while the pattern comprises representative questions which gets matched against the user queries. The response tag contains the sample answers to the user query related to the associated tag. The context_set and context_filter are used for enabling context of the conversation. The intent.json had 150 classes, i.e., 150 questions to answer pairs for the model to be trained upon. Figure 4 comprehensively outlines the sequential steps followed to preprocess the raw healthcare question answer data.

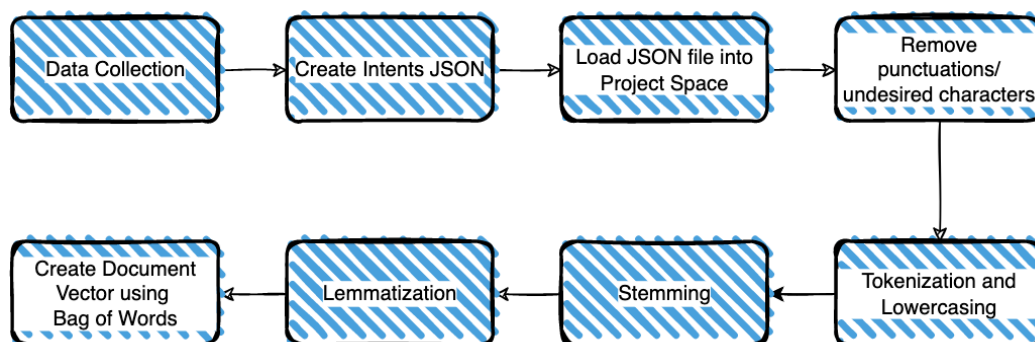


Figure 4: Data Preprocessing

At first, the gathered up medical question answer pairs were stored in JSON format and loaded into project space using json package. Then, using regular expression, the undesired characters and punctuations were removed from the dataset. Following that, the individual patterns were broken into tokens using nltk.word_tokenize and then tokens were converted into lowercase for uniformity. PorterStemmer package was utilized to strip the last few characters of the words therefore reducing the inflection in the tokens and afterwards, WordNetLemmatizer was incorporated to convert the stemmed tokens into lemmas in order to bring the tokens to their base form. The primary purpose of applying stemming and lemmatization was to normalize the texts and achieve dimensionality reduction. Once the raw text are converted to lemmas, a document vector was created using Bag of Words technique. Figure 5 shows a sample example how a raw sentence is processed through various NLP techniques in the proposed solution.

The BoW was performed as the machine learning models are unable to operate directly with texts, and they must be transformed into numbers, vectors of numbers to be precise. In the incorporated BoW technique, first an empty array with length equal to the total number of words was created. Then for every word in the current document is matched

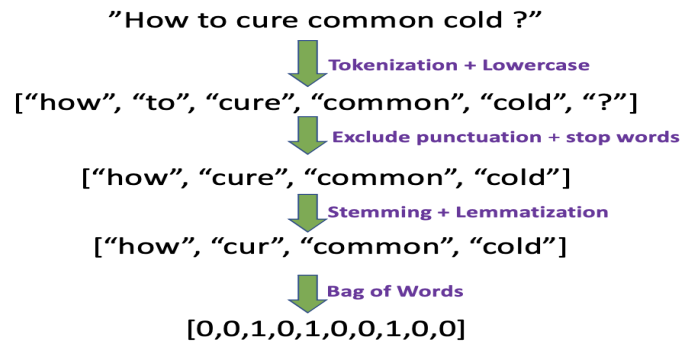


Figure 5: Sample of NLP pre-processing step

against all words, and if present, 1 is inserted in the bag array. The BoW technique is distinctly illustrated in the below diagram 6.

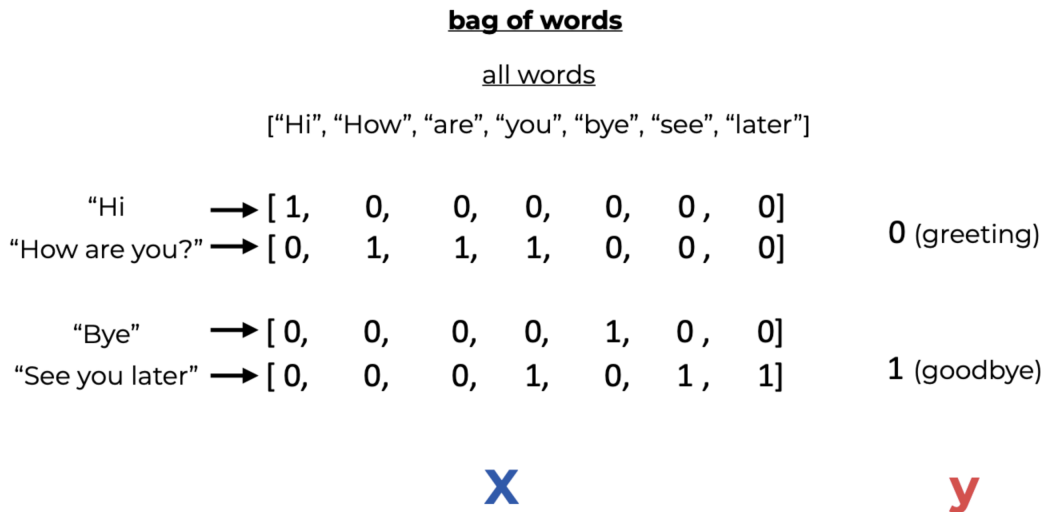


Figure 6: Bag of Words Technique

5.3 Architecture of the Deep Neural Network Model

For training the pre-processed dataset, a Sequential multi-layered deep neural network was developed using keras library. Keras is a sophisticated and easy-to-use open source software Python framework for building and analysing deep learning models. It combines Theano and TensorFlow, two effective and convenient mathematical computing frameworks, to make it easier to create and train neural network models. For implementing the DNN model, blocks like input and output Dense layer, dropout layer were created. Also, activation functions like ReLU and Softmax were used at appropriate layers.

The developed deep neural network has four dense layers with one input layer, one output layer and two hidden layers. The first layer/input layer has 128 neurons with input shape same as the length of the training data. Rectified Linear Unit (ReLU) activation function is included in this layer. Activation function in a neural network is necessary to convert the node's accumulated weighted input into the node's activation or output for that input. ReLU is a linear function, which outputs the input directly if the input value is positive, else it transmits zero. There is a risk of data overfitting when training a big

neural network with a relatively small input dataset, hence resulting in poor performance while evaluating on fresh data. To mitigate this overfitting issue, one approach is to randomly drop out some neurons. Therefore, after the input dense layer, a dropout layer with rate 0.5 was used to prevent over-fitting. To simplify, this drop out layer with rate 0.5 randomly excludes 50% of the inputs in each update iteration. The second dense layer was a hidden layer comprised of 64 neurons, ReLU activation function and followed by a dropout layer with rate=0.5. The third dense layer was also a hidden layer with similar configuration to the previous intermittent hidden layer. The final layer was the output layer with shape equal to the number of tags/classes in the training dataset. the figure 7 illustrates the architecture of the DNN model.

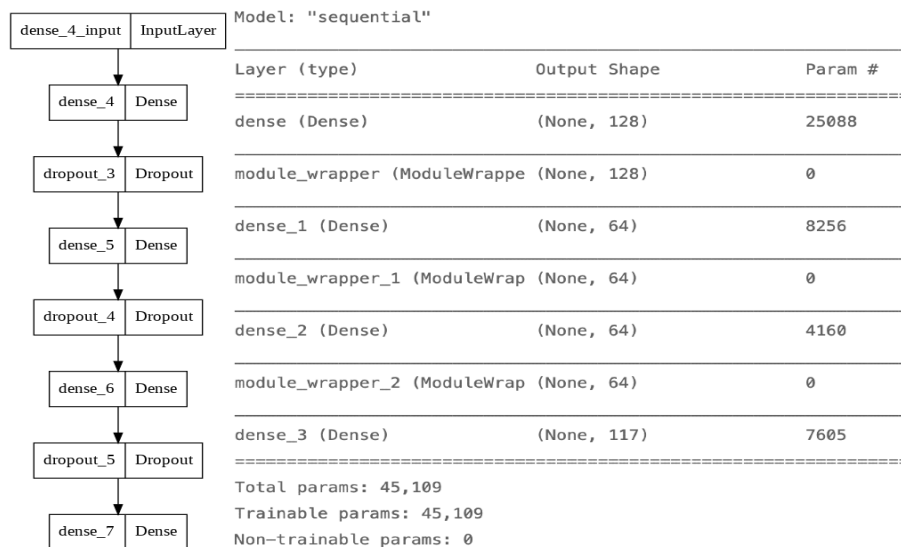


Figure 7: DNN Architecture

5.4 Enabling Contextualization in the Model

The proposed solution was able to remember the context of the ongoing conversation. For example, if the user had already asked “how to treat Fever?”, in the next query if the user wants to know the symptoms of Fever, they need not to ask “What are the symptoms of fever?”, rather they can just type in “symptoms” or “what are the symptoms?” without naming the disease name and the chatbot could provide the correct symptoms of Fever. The contextualization capability can also be implemented to ask other questions like diagnosis, causes and risk factors etc without naming the disease name in the conversation again.

To enable this feature, “state” was added to the model. The state was enabled by implementing a python “Dictionary” data structure which holds the context of the conversation. In the intents json file, each intent has a tag called “context_set” or “context_filter”. While iterating through each intent, the model finds out the value of the context_set and set it as the Context. In the next user query, the model check the value of the context_filter and if found the value in context_filter matching with the value of Context, then the model replies answers only from that particular intent. The same logic is explained in detailed in the below pseudocodes diagram 8.


```

{"tag": "Fever",
 "patterns": ["How do you treat a mild Fever?", "what to do if i get a mild fever?",
 fever?", "fever"],
 "responses": ["To treat a fever at home: 1)Drink plenty of fluids to stay hydrated.
 light blanket if you feel chilled, until the chills end. 4)Take acetaminophen (Tyle
 others). 5) Get medical help if the fever lasts more than five days in a row."],
 "context_set": "fever"
},

{"tag": "Symptoms",
 "patterns": ["what are it's symptoms", "symptoms of cold", "symptoms"],
 "responses": ["Symptoms of fever are : 1)Temperature rise 2)Headache 3)Body pain"],
 "context_filter": "fever"
},

```

```

For intent in intents:
    if context_set is not Null:
        context = context_set.value

    if (context_filter is not Null and context_filter == context):
        print(intent[responses])
    else:
        context = Null

```

Figure 8: Pseudocode for applying contextualization

5.5 Architecture of the Naive Bayes Model

To compare the performance of the deep neural network model, a Naive Bayes model was also developed, and then the end results of both the models were weighted up. Among various machine learning algorithms, Naive Bayes is presumed to be the best classifier while dealing with sparse matrix data. The reason for this is that sparse data contains many zeros/missing values, and the time and space complexity of the Naive Bayes method grows nicely with the amount of non-missing values. The Naive Bayes algorithm uses the Bayes theorem with a Naive assumption of no association between distinct features. As per the Bayes theorem:

$$Posterior = likelihood * proposition / evidence$$

or

$$P(A|B) = P(B|A) * P(A) / P(B)$$

For implementing the Naive Bayes model, GaussianNB was imported from sklearn.naive_bayes package. While fitting the training data, default parameters were used in the GaussianNB classifier.

6 Evaluation

Following the successful creation and implementation of the models, outcomes of the models were evaluated with numerous accuracy metrics. It is essential to determine whether the outputs are as expected, as this estimation is regarded critical since it allows a comparison to be made between the projected and true results. In this section, performance and efficiency of the two models i.e., Deep Neural Network and Naive Bayes were analysed and compared. For assessing the performance of the models, various metrics like Training Accuracy, Training Loss, Validation Accuracy, Validation Loss, f1 score, Precision, and Recall measures were exercised.

6.1 Deep Neural Network Model Evaluation

The deep neural network was executed with 200 epochs as shown in the figure 9. The training dataset (input sparse matrix) of the model included 1312 rows and 310 columns

as independent variables, whereas the dependent variable was a sparse matrix with 1312 rows and 196 columns. Similarly, the validation dataset had 119 entries. Diagram 10 illustrates the accuracy and loss of the DNN model while being trained and evaluated with training and validation dataset. From the Accuracy vs Epoch plot, it is quite evident that the training accuracy of the model steadily improved until the 18th epoch and afterwards remain became constant at around 0.97. Although the validation accuracy showed a similar trend, it has remained fluctuating between 0.7 and 0.8 after the early epochs. Similarly, the training loss was sharply declined from 3.8 to 0.2 between 1st epoch and 17th epoch and then remained constant at 0.19. The validation loss followed a more turbulent pattern, with loss value oscillating between 2 and 3 after the 25th epoch. This contrast between the training and validation data performance can be explained by the size of the dataset. As the overall dataset is quite limited and the variation in the validation dataset is much wider, the training accuracy/loss varied largely from validation accuracy/loss. The DNN model achieved F1 score of 75%, Precision of 83%, and Recall score of 74%. As the training data is a sparse data with multiclass dependent value, the F1 score, Precision, and Sensitivity were computed with weighted average. The weighted average calculates the performance of each class, by considering the number of instances present in that class.

```

Epoch 195/200
1380/1380 [=====] - 0s 214us/sample - loss: 0.0413 - acc: 0.9732 - val_loss: 1.4566 - val_ac
c: 0.8319
Epoch 196/200
1380/1380 [=====] - 0s 212us/sample - loss: 0.0398 - acc: 0.9768 - val_loss: 1.4653 - val_ac
c: 0.8230
Epoch 197/200
1380/1380 [=====] - 0s 212us/sample - loss: 0.0403 - acc: 0.9746 - val_loss: 1.4548 - val_ac
c: 0.8142
Epoch 198/200
1380/1380 [=====] - 0s 215us/sample - loss: 0.0402 - acc: 0.9761 - val_loss: 1.4562 - val_ac
c: 0.8496
Epoch 199/200
1380/1380 [=====] - 0s 213us/sample - loss: 0.0399 - acc: 0.9746 - val_loss: 1.5458 - val_ac
c: 0.8407
Epoch 200/200
1380/1380 [=====] - 0s 211us/sample - loss: 0.0406 - acc: 0.9761 - val_loss: 1.4374 - val_ac
c: 0.8584
model created

```

Figure 9: Total no. Epochs with Training and Validation Accuracy/Loss

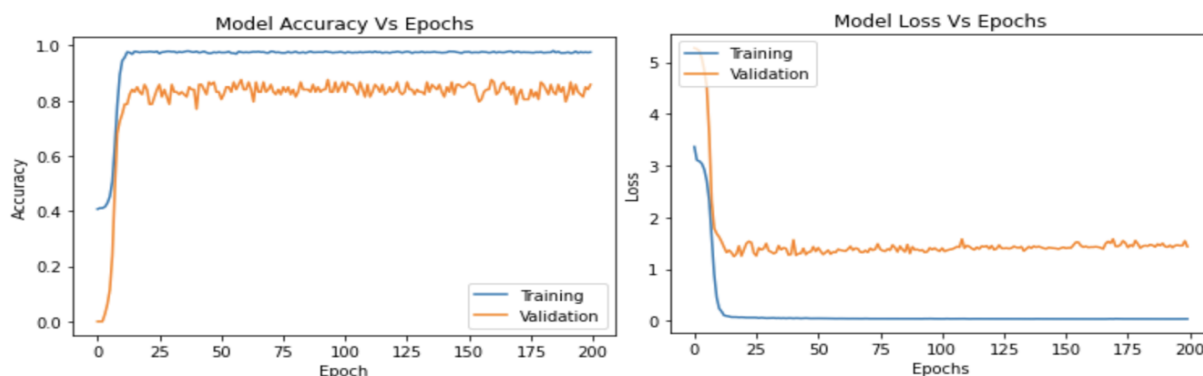


Figure 10: Accuracy, Loss Vs Epoch Graph

6.2 Evaluating the Deep Neural Network Model Output by Testing with Real-time User Queries

After the Deep Neural Network was trained with input data, the model was feed with real-time user queries to check the performance of the proposed solution. Figure 11 shows sample model response while applied with live user questions. Firstly, the user typed in “hello”, to which the chatbot successfully identified it as part of the greetings tag and replied with the sample response “Hi There !!”. Then the user asked “How to cure common cold?”, to which the chatbot replied with the step-by-step procedure/medicines to take in case of common cold. Here, it can be observed that the model successfully identified the tag “Cold” from the user input and replied with the appropriate answer. In the third query, the user just asked “what are its symptoms” without naming the disease name, but the model already remembered the context of the conversation i.e., the disease name from the previous question and was able to reply the correct symptoms of common cold. A user defined threshold of 0.65 is set, so that any predictions with probability greater than 65% is only considered as a valid response and shown to the user.

```
In [*]: chat()
execution queued 16:31:17 2021-12-14

Start Talking with the bot(type quit/q/bye to stop!)
You: hello
bot: Hi There!!

You: How to cure common cold?
bot: 1)Keeping hydrated is absolutely vital to help 'flush' out the cold, as well as to break down congestion and keep your throat lubricated. 2)Vitamin C is extremely helpful when fighting infection, so at the first sign of a cold be sure to increase your intake by eating plenty of berries, citrus fruits, papayas, broccoli and red peppers which will help keep you protected. 3)When it comes to combating a cold,Vitamin D is essential in helping to regulate immune response.

You: what are its symptoms..
bot: 1. blocked or runny nose 2. sore throat 3. headaches 4. muscle aches 5. coughs 6.sneezing 7. a raised temperature 8. pressure in your ears and face 9. loss of taste and smell

You: what medicines to take incase of fever?
bot: To treat a fever at home: 1)Drink plenty of fluids to stay hydrated. 2)Dress in lightweight clothing. 3)Use a light blanket if you feel chilled, until the chills end. 4)Take acetaminophen (Tylenol, others) or ibuprofen (Advil, Motrin IB, others). 5) Get medical help if the fever lasts more than five days in a row.

You: its symptoms?
bot: Symptoms of fever are : 1)Temperature rise 2)Headache 3)Body pain
```

Figure 11: Solution’s response to real-time queries

Similarly, in the consequent questions, the chatbot was successfully able to answer the treatment process and symptoms of Fever. From these two sample examples, it can be perceived that the proposed model is effectively able to answer the user queries and able to remember the context of the conversation.

6.3 Comparing the DNN model performance with the Naive Bayes Model

The below table 2 outlines and compares various evaluation metric score between the Deep Neural Network model (DNN) and the Naive Bayes model (NB). The training accuracy of the DNN model has achieved an exceptional 98.62%, while the Naive Bayes model also attained 97.9% accuracy score. The test accuracy of the Neural Network model was around 81.42%, while the Naive Bayes model accomplished a relatively lesser

77.87% test accuracy. Likewise, the DNN model also recorded better F1 score, Precision, and Recall score than that of the Naive Bayes model.

Table 2: Performance comparison between DNN and NB model

Evaluation Measure	Deep Neural Network Model	Naive Bayes Model
Train Accuracy	98.62%	97.9%
Test Accuracy	81.42%	77.87%
F1 Score	83.0%	80.5%
Precision	90.0%	88.5%
Recall	81.0%	77.9%

From the above comparison, it can be acknowledged that the deep neural network model was more efficient in generalizing and classifying the input training data, hence yielding a better accuracy score. Therefore, it can be inferred that, though the Naive Bayes algorithm performs remarkably with text input (sparse data), but by equipped with features like multiple hidden layers, back propagation and appropriate weight initialization, the deep neural network outperforms it.

6.4 Discussion

The primary area of focus in this study was to enable contextualization in the chatbot, while answering user queries with appropriate predefined responses. From validating the model with real-time queries, it can be reckoned that the model was successfully able to remember the context of the conversation i.e., the disease name from the previous user query and able to provide the symptoms in the subsequent query without asking the disease name again. Coming to the training accuracy, the model has an exceptional 98% accuracy score, which infers that the DNN model was successfully able to classify the training variables. On the other hand, the validation accuracy achieved a decent 81.42% accuracy score, which indicates that the model struggled a little while classifying with new data. This behaviour can be explained by the size of the training dataset and the diversity in the validation dataset. The training and validation loss also followed the similar trend as of training/validation accuracy. The model attained a fair 81% of weighted F1 score, indicating that the solution performs decently with unbalanced dataset. The weighted Precision and Recall score confirmed that the model was able to identify the true positives efficiently. The model was run with 200 epochs, but the accuracy and loss value became stagnant after early epochs (25th epoch). Hence, it can be concluded that running the model with 200 epochs is not necessary and executing until 50 epochs would be sufficient.

On comparing the proposed model’s performance with other state-of-the art models, it can be observed that Makhkamova et al. (2020) achieved a training accuracy of 90% but more importantly, their solution achieved an impressive 80% of validation accuracy. Also, the study conducted by Dasanayaka and Warnajith (2020) used LSTM and RNN to achieve a magnificent 90% of validation accuracy while able to identify and response to non-English Sinhala language text. The contextual chatbot developed by Keerthana et al. (2021) attained a validation accuracy of 88% by using Deep Reinforcement Learning techniques and using a much larger training dataset. All the above distinguished studies indicates that by using a larger and diverse dataset, the validation accuracy can be improved significantly. The table 3 outlines the detailed comparison of train and validation

accuracy between the proposed model and other state-of-the-art models.

Table 3: Performance comparison with the State-of-the-art Models

Authors	Methods Used	Training Accuracy	Validation Accuracy
Proposed Model	Deep Neural Network	98.62%	81.42%
Makhkamova et al. (2020)	Bi-LSTM, CNN	90%	80%
Dasanayaka and Warnajith (2020)	LSTM, RNN	75.0%	58.2%
Keerthana et al. (2021)	NMT, BiRNN, DR-DQN	88%	80%
Khadilkar et al. (2021)	Convolutional Transformer	87%	72%

7 Conclusion and Future Work

Artificial Intelligence has offered a lot of dimensions and capabilities in the field of healthcare in the form of virtual assistance and conversational systems. Still, there is room for development in this sector in terms of accuracy and relevance, hence the goal of this research is to increase model accuracy and provide contextualization to chatbot solutions. To achieve the defined research objectives of this study, several state-of-the-art conversational system research articles were extensively studied and analysed, and then a multi-layered deep neural network model was constructed to be trained on the preprocessed input dataset. The healthcare related question-answer data was gathered from numerous open-source websites and then stored in JSON format. The raw input data was then cleaned and transformed using several NLP techniques like tokenization, stemming, lemmatization followed by a Bag of Words (BoW) approach for feature extraction and word embedding. The output of BoW, a word vector, was then fed into a four layered deep neural network with ReLU activation function in the intermittent layers and Softmax activation function in the output layer for training purpose. To enable contextualization, a python dictionary data structure was developed using `context_set` and `context_filter` mechanism. A Naive Bayes model was also created and trained to compare its results with the DNN model. For evaluating the performance of the model, metrics like Accuracy, Loss, F1 Score, Precision, and Recall were used. The Deep Neural Network model attained a training accuracy of 98% and validation accuracy of 81%.

From the accuracy score, it can be observed that while the training accuracy score is quite overwhelming, the model struggled a bit with fresh data. This tendency can be explained by the fact that the input data size is relatively smaller than a real-world production project. Therefore, in the future work, a comparatively larger and diverse dataset can be utilized to create the input knowledge base. Furthermore, for feature extraction and word vector creation, more advanced techniques like TF-IDF, Counter Vectorizer can be utilized. Also, for training the input data, a more advanced deep learning model using Deep Reinforcement Learning can be implemented to increase the accuracy and enable better contextualization.

8 Acknowledgment

I sincerely appreciate and express my deep gratitude to Dr. Majid Latifi, my thesis supervisor, for his unwavering, round-the-clock assistance and direction in finishing my research project. During the scheduled meetings, Dr. Latifi assisted, examined, and offered input on my project report and code, which helped me substantially to complete my research project.

References

- Astuti, W., Putri, D. P. I., Wibawa, A. P., Salim, Y., Purnawansyah and Ghosh, A. (2021). Predicting frequently asked questions (faqs) on the covid-19 chatbot using the diet classifier, *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021* pp. 25–29.
- Carranza, K. A. L. R., Manalili, J., Bugtai, N. T. and Baldovino, R. G. (2019). Expression tracking with opencv deep learning for a development of emotionally aware chatbots, *2019 7th International Conference on Robot Intelligence Technology and Applications, RiTA 2019* pp. 160–163.
- Dasanayaka, D. D. and Warnajith, N. (2020). Contextual assistant framework for the sinhala language, *Proceedings - International Research Conference on Smart Computing and Systems Engineering, SCSE 2020* pp. 45–50.
- Deepika, K., Tilekya, V., Mamatha, J. and Subetha, T. (2020). Jollity chatbot- a contextual ai assistant, *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020* pp. 1196–1200.
- Dutt, V., Sasubilli, S. M. and Yerrapati, A. E. (2020). Dynamic information retrieval with chatbots: A review of artificial intelligence methodology, *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1299–1303.
- Gupta, J., Singh, V. and Kumar, I. (2021). Florence- a health care chatbot, *2021 7th International Conference on Advanced Computing and Communication Systems (IC-ACCS)*, Vol. 1, pp. 504–508.
- Kandpal, P., Jasnani, K., Raut, R. and Bhorge, S. (2020). Contextual chatbot for health-care purposes (using deep learning), *Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4 2020* pp. 625–634.
- Keerthana, R. R., Fathima, G. and Florence, L. (2021). Evaluating the performance of various deep reinforcement learning algorithms for a conversational chatbot, *2021 2nd International Conference for Emerging Technology, INCET 2021* .
- Khadilkar, A., Patil, H. Y. and Sundaramali (2021). Context sentences to single vector compression using convolutional transformers for deep learning based nlg tasks, *2021 International Conference on Computer Communication and Informatics, ICCCI 2021* .

- Makhkamova, O., Lee, K. H., Do, K. and Kim, D. (2020). Deep learning-based multi-chatbot broker for qa improvement of video tutoring assistant, *Proceedings - 2020 IEEE International Conference on Big Data and Smart Computing, BigComp 2020* pp. 221–224.
- Nursetyo, A., Setiadi, D. R. I. M. and Subhiyakto, E. R. (2018). Smart chatbot system for e-commerce assistance based on aiml, *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 641–645.
- Prasetyo, A. and Santoso, H. A. (2021). Intents categorization for chatbot development using recurrent neural network (rnn) learning, *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021* pp. 551–556.
- Quan, T., Trinh, T., Ngo, D., Pham, H., Hoang, L., Hoang, H., Thai, T., Vo, P., Pham, D. and Mai, T. (2019). Lead engagement by automated real estate chatbot, *NICS 2018 - Proceedings of 2018 5th NAFOSTED Conference on Information and Computer Science* pp. 357–359.
- Raundale, P. and Sawale, A. (2021). Dialog prediction in institute admission: A deep learning way, *2021 2nd International Conference for Emerging Technology, INCET 2021* .
- Sriram, S., Manikandan, J., Hemalatha, P. and Roselin, G. L. (2021). A chatbot mobile quarantine app for stress relief, *2021 International Conference on System, Computation, Automation and Networking, ICSCAN 2021* .
- Tran, Q. D. L. and Le, A. C. (2021). A deep reinforcement learning model using long contexts for chatbots, *Proceedings of 2021 International Conference on System Science and Engineering, ICSSE 2021* pp. 83–87.
- V, M., Manur, D., Farook, M. F., K S, C. G. and Desai, S. M. (2021). Doctor patient assistance system using artificial intelligence, *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 534–538.
- Vailshery, L. S. (2021). Number of voice assistants in use worldwide 2019-2024(in billions), *Statista, Technology and Telecommunications, Consumer Electronics, April 2020* .
- Yin, J., Chen, Z., Zhou, K. and Yu, C. (2019). A deep learning based chatbot for campus psychological therapy, *arXiv preprint arXiv:1910.06707* .