

Contextual Chatbot for Healthcare Purposes (using Deep Learning)

Prathamesh Kandpal
Dept. of Electronics &
Telecommunication
Vishwakarma Institute of
Technology
Pune, India
prathamesh.kandpal16@vit.edu

Kapil Jasnani
Dept. of Electronics &
Telecommunication
Vishwakarma Institute of
Technology
Pune, India
kapil.jasnani16@vit.edu

Ritesh Raut
Dept. of Electronics &
Telecommunication
Vishwakarma Institute of
Technology
Pune, India
ritesh.raut16@vit.edu

Prof. Dr. Siddharth Bhorge
Dept. of Electronics &
Telecommunication
Vishwakarma Institute of
Technology
Pune, India
siddharth.bhorge@vit.edu

Abstract—As the demand in Machine Learning & AI keeps growing, new technologies will keep coming in the market which will impact our day-to-day activities, and one such technology is Virtual Assistant Bots or simply Chatbots. Chatbots have evolved from being Menu/Button based, to Keywords based and now Contextual based. The most advanced among all of the above is contextual based because it uses Machine Learning and Artificial Intelligence techniques to store and process the training models which help the chatbot to give better and appropriate response when user asks domain specific questions to the bot. In this paper we will be not only discussing about the working of our model but also the applications and relevant work conducted in this domain, also there will be discussion about the challenges and future scope of this technology. For this work, neural networks have been used to train data and various packages which help us in giving better results. In this chatbot we will be integrating the concepts of Natural Language Processing with Deep Learning for getting better results. Healthcare plays a wide role in our daily lives, whenever a person feels sick he/she visits their family doctor or any nearby clinic just to get to know what issues they are facing, in the recent years many companies & institutions have collaborated with hospitals to provide support which can help doctors and medical staff to deal with patients in better way and reduce their efforts with the help of technology. Chatbots can play a major role in reshaping the healthcare industry by providing either predictive diagnosis or any other assistantships like booking an appointment.

Keywords—Neural Networks, Deep Learning, Contextual Chatbot, Artificial Intelligence, Machine Learning, Healthcare.

I. INTRODUCTION

To being with the definition of chatbot, Chatbots are software applications which help in simulating human like conversations through voice commands or text-based chats or both. Its, a service backed by rules and in some cases Artificial Intelligence which converses with the customer through different platforms of communication. Chatbots constantly require training and testing to fit with the ideal standards set by the developers. The purpose for taking such kind of task was to understand the importance of virtual assistants in the coming years & how they will be responsible for market disruption in many fields of businesses. They have many applications in the domain of customer service, request routing or information gathering. A research by Gartner Inc. predicts that by the end of 2020, 25% of consumer services & support operations will be using Virtual Customer Assistant (VCA) technologies including chatbots. It has also been reported that there has been 70% reduction in calls, chats, and emails with virtual assistantship [21]. So, there is a huge scope and requirement in these domains. Tech-giants have already invested in this

domain and have come up with products like Amazon Lex, Google Cloud's Dialog flow, Microsoft Azure Bot.

For this work we have combined the concepts of TensorFlow, TFLearn, NLTK & NumPy with the field of healthcare assistance. In recent years Healthcare organizations with the help of educational institutions & tech-driven businesses have collaborated on lot of projects to build a feasible platform for all needs but such work requires a lot of cost, time, effort, collaboration and negotiation. Our approach is much simpler and such work can be used by any large scale or small-scale implementation; it all depends upon the purpose of use. The program developed in this work is compatible with any domain, but it has been specifically developed for healthcare purposes. We have used intents to train our model, these intents consist keywords, responses & patterns. Ahead during the implementation part there will be detailed explanation on this part. Also, there will be discussion on applications and challenges.

This chatbot is an attempt to let users understand the symptoms they are facing and get a basic diagnosis about the diseases they could be having. The diagnosis given by bot may or may not be accurate. For accurate diagnosis and necessary medication, patients should always visit the doctor. Also, the patient should visit a doctor if the symptoms get worse or don't improve. It is important to understand that the chatbot developed for this work is not designed to give actual or authorized diagnosis of a disease.

The goal of chatbot is to give an overall idea regarding the kind of disease the individual might be facing it can be true or false. Virtual Assistants making a conversation via textual methods are contributing to identifying symptoms, chronic health problems managing medications and monitoring. Smartphone usage with the rising adoption of health applications, IoT, telehealth and other similar technologies are contributing to the market growth in the healthcare industry. Things like virtual assistants essentially save time and reduces burden on doctors. Major benefit of using chatbots is that customer can ask any query without knowing the proper keywords, the machines easy interpret the natural language by comparing the words related to that query and present accurate responses without any hassle. Just like the search engines, bots work in the same fashion [1].

This technology is evolving each day, developers & researchers are working on newer ways for bots to understand humans better, give appropriate answers, predict the next query & store the previous conversations.

II. APPLICATIONS OF CHATBOT IN VARIOUS DOMAINS:

A. *Messaging Applications*

In modern day, many companies have integrated their usual business services on messaging apps like WhatsApp or Facebook Messenger. They are used for Business to Customer relations like sales, marketing or simply customer service. In 2016, Facebook allowed developers to use messenger for bot purposes as it accounts to billions of users and it can be a great platform for businesses to connect with their audience. More than thirty thousand bots were developed by many businesses and developers using Messenger as a platform within 6 months of its release, this number rose to more than 100,000 by the end of September 2017. Some Airlines like KLM & Aeroméxico had launched customer services on the Facebook Messenger platform even before Facebook launched messenger as a platform for developers and businesses to use it for virtual assistantship purposes and currently announced their participation in the testing in newer virtual assistant services. Many businesses like banks, media companies, health care providers, retail businesses, hotel and restaurant chains, e-commerce websites and government bodies are currently using chatbots for handling easy and simple queries to enhance customer engagement. According to a 2017 study chatbots were so insignificant that only 4% of organizations used them. But a 2016 study also showed that 80% intended to use them by 2020 [3].

B. *As a part of Organizations Applications & Websites*

In India many financial banks and institutions have integrated chatbots on their website portals for customer to transact easily and safely. An e-commerce company named Rare Carat used IBM Watson Assistant as a platform to provide information to prospective diamond buyers which is better than previous technologies [3].

C. *Customer Services*

As the demand for chatbots and virtual assistants keep growing, the demand for better customer engagement will keep increasing. Major banking and fin-tech organizations are planning to integrate chatbots for advancing customer services cheaper and faster assistance to their clients. In modern day, chatbots are capable of efficiently conducting a dialogue and replacing other communication handles like phone, SMS, or email [3].

D. *Politics*

Semantic Analysis Machine or simply SAM is a chatbot created by Nick Gerritsen is a New Zealand based chatbot which can share political opinions on topics like healthcare, climate change, education and more [3].

E. *Toys*

ToyTalk is a company which collaborated with Barbie toys to create an internet connected doll (Hello Barbie) which is integrated with inbuilt functionalities of a chatbot. ToyTalk had previously created range of smartphone-based characters for kids' recreational activities. The behaviours of these characters are set by a certain parameter and they are constricted to perform conversations which are built specifically for their characters, this leads to a different

storyline for each character. There are also certain toys which are built for educational purposes like CogniToys Dino. IBM's Watson Computer has played an integral part for such educational chatbots. Many of these toys use speech recognition as a basic tool for conversing with kids [3].

F. *Platforms for internal use*

Some companies are exploring different methods in which they can integrate chatbots across various departments internally, like Customer support, HR related queries or IoT based projects. Mila is a chatbot developed by Overstock.com which aims to automate simple yet time consuming processes like applying for sick leaves. Some other large companies like Royal Bank of Scotland, Lloyds Banking Group, Renaults, and many more are using chatbots in internal departments rather than using call centres as the medium for getting first point contact. Since the F8 Conference, the ecosystem around the SaaS based chatbots are steady changing and growing as it marked an important moment because Facebook then allowed messenger as a platform for using chatbots. Many IT Architects are designing intelligent chatbots which can share knowledge and unlocking useful experiences inside a company much easily, this has also helped in reducing errors from viable service desks significantly. Intelligent chatbots are able to perform easy as well as some difficult queries because they are powered with modern day technologies like Machine Learning, Natural Language Generation, Natural language Understanding, Deep Learning and many more. [3].

G. *Chatbot Sequences*

Chatbot Sequences are script sequence of messages used by marketers which work just like Autoresponder sequences. These sequences are produced by user opt-in or use of targeted keywords which users use while making interactions. The concept of Decision Tree is used for building strong chatbot sequences so that whenever a user asks a query, the decision tree navigates the system to correct responses and returns it to the end-user's interface [3].

III. RELEVANT WORK

Chatbots have been in existence since 1966 and many contributions have been made to this field. Earlier chatbots never used Natural Language for communications like speech or text. In the current day and age chatbots are getting very famous and big corporations like Google, Microsoft and Amazon are investing in this field to build frameworks which other businesses can use for their own purposes like customer care or ordering something online. Amazon Alexa is a great example of Virtual Assistantship as it provides help in various day-to-day activities like setting alarms, listening to music, news updates and online shopping just by speaking.

Below is a list of the widely available frameworks and services offered by enterprises and recent development in past couple of years. The modern day chatbots use the concepts of Natural Language Understanding and Deep Learning functionalities like automatic speech recognition (ASR).

A. Dialogflow by Google

Dialogflow is a tool which helps in conversational user interface for mobile and web applications. Currently Dialogflow is a part of Google Cloud platform which helps businesses in providing latest tech services related to the field of Cloud Computing, Databases, Machine Learning and AI. Dialogflow can analyze both voice and text-based user inputs. Dialogflow can be integrated on whichever platform the developer wants to deploy and it has the ability to interact with various devices like mobile phones, car, TV, Google home or phone calls [16]. For asking any information to a chatbot there can be multiple possibilities and hard coding all these possibilities can be time consuming, to tackle with this issue the Dialogflow uses the concepts of Natural Language Understanding (NLU). NLU helps the in converting the human language into the computer language and vice versa. NLU is just like natural language processing but it goes a bit further to understand terminologies like sentiment analysis, accents, summarizations and understanding questions on which it has never been trained before.

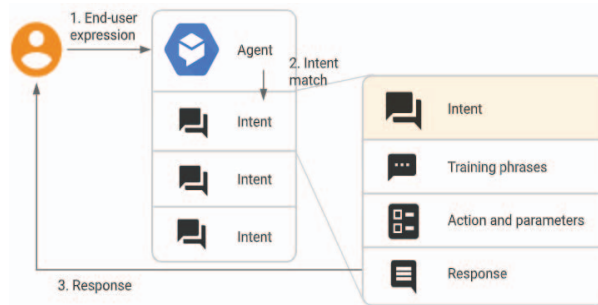


Fig 1: Dialogflow Algorithm

Dialogflow uses virtual agent which helps in handling conversations with the user, once the user gives trigger to this model it starts finding the related intent, intents are the trained conversations setup by the developer. Each intent consists of training phrases, actions & parameters, and responses. Once the correct intent is mapped, the chatbot returns a response to the end-user's query [22].

B. Amazon Lex

Amazon Lex is an AWS service which is used for building conversational interfaces for both text-based & voice-based inputs. This service started back in April 2017 and it originally powered Amazon Alexa virtual assistant. Developers used Alexa Voice Service as a platform for Alexa based service but recently Amazon Lex is used for both voice & text-based purposes also its convenient to handle with databases because its integrated with AWS. Just like Dialogflow, Amazon Lex also uses NLU & ASR but there is another concept called Speech Language System (SLU) which they have implemented in their framework, through SLU, the chatbot takes in responses learns the actual meaning of users input and fulfils the user purpose by invoking applicable business actions. Amazon Lex is powered by pre-built integration with AWS Lambda which provides bots access to pre-built serverless enterprise to link data in SaaS applications like Salesforce, HubSpot or Marketo [20].



Fig 2: Sample architecture of Informational Chatbot (OhioHealth)

C. Microsoft Azure Bot Service

Microsoft Azure provides Bot Framework Service, and it can be developed in multiple languages like Python, JavaScript, .NET or C#. Azure Bot has six stages for any enterprise or freelance developer for achieving their goal, they include Planning, Building, Testing, Publishing, Connecting and Evaluating. This framework helps developers with many design guidelines for planning the architecture of the final product. Some of the key features provided by this bot framework are adding natural language processing, adding knowledge base which will help in returning efficient replies, managing multiple models like simple script based & contextual based and adding graphics, cards and buttons to enhance the user experience. Azure Bot can be used on various platforms like Facebook Messenger, Kik, Slack, Microsoft Teams, Telegram, SMSs, Twilio & Cortona. HTTP POST request is used for sending and receiving messages from the end-user [19].

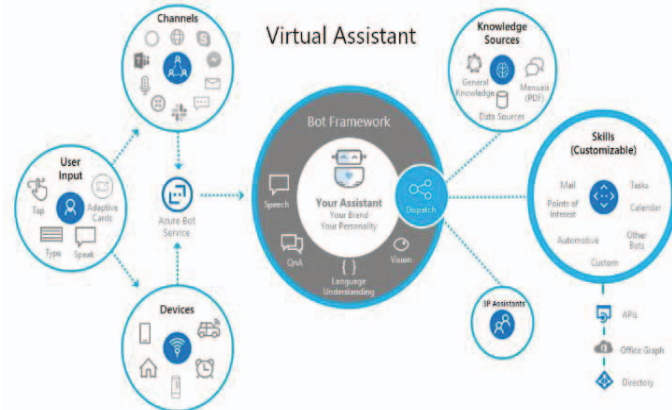


Fig 3: Azure Bot Implementation Workflow

D. Watson Assistant by IBM

Watson Assistant is a service provided by IBM Cloud which uses REST APIs. This makes them compatible as a solution within an application. Which means they can be integrated on various platforms like Slack, Facebook Messenger, WordPress, or any Custom APIs on which the consumer can interact with Watson [16]. IBM Watson Assistant has been recognized for its ease of use because it provides good and easy to navigate user interface. The

assistant gets end-user query and routes it to the dialog skill. The dialog skill interprets and understands the user input further which helps it in directing the flow of the conversation. The dialog gathers all necessary information it needs to respond or perform a request on the user's behalf. Questions which cannot be answered by the dialog skill are sent to the search skill, this helps in finding relevant answers by searching the company knowledge bases that developers configure for their own purposes [10].

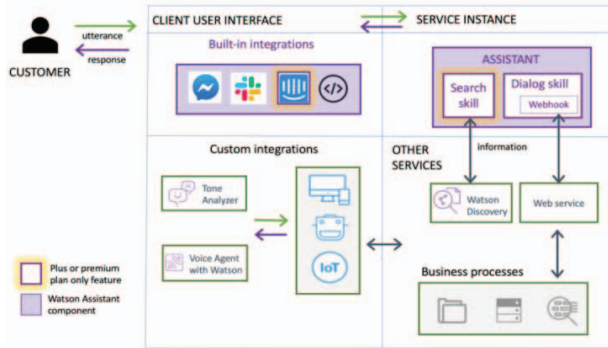


Fig 4: IBM Watson Implementation

E. Facebook Messenger Bot

In April 2016, Facebook released its bot platform on Messenger, this service included APIs to build chatbots which would be capable for doing meaningful interactions with the end user, they have since made many enhancements for providing better experience: [15]

- **Virtual Assistants in group chats:**

Bots didn't talk like a casual member on but they did by notifying day-to-day news updates, weather updates, sports news, payment receipts and many more.

- **Chat extensions:**

Facebook launched Discovery tab which features variety of bots used, different categories of bots, recent trends and searching ability. Facebook users can also play games, book flight tickets, collaborate for music playlist while interacting with specific apps

- **QR Scan:**

Brands have also collaborated with Facebook for creating bots for helping customers with necessary queries, users can scan the QR code and directly interact with the bots.

The Facebook messenger bot works by triggering Webhooks on the business server's URL from where the messaging services are hosted. Then, Send API helps in conversing with the users. By using this technique, Developers can build well-constructed conversations to which will guide users in a proper flow of conversations and help them achieve their purpose of using the developers chatbot [14].

F. RASA – Open Source conversational AI

Rasa provides infrastructure & tools necessary for high-performing, resilient, proprietary contextual assistants that work. Developers can create better text- and voice-based assistants using RASA. RASA is an open source platform which uses machine learning framework as the basic building block for creating automated texts, voice-based conversations. RASA provides the functionality of holding

conversations. Connecting to different messaging channels and with the APIs. RASA being an open source helps even in migration from other platforms like Dialogflow, Wit.ai, LIUS and IBM Watson [11]. RASA NLU helps in intent classification, response retrieval and entity extraction in chatbots and RASA Core is used for holding conversations and decides what to do next. RASA Core uses machine learning model instead of using many if/else statements to decide what next actions can be performed. For training the data and creating instances, developer can either use Markdown format or JSON format [6]. For working on this platform, one needs to be familiar with Python and installing packages, even if there is no familiar background to this field, developers can easily create chatbots by following step-by-step guide provided on their website.

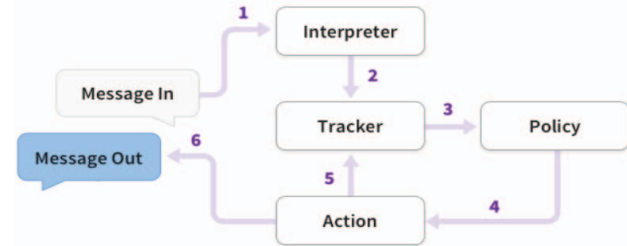


Figure 5: RASA's Message Handling Technique

G. Sequence to Sequence

Seq2Seq uses Recurrent Neural Networks which is specially designed Deep Neural Architecture for NLP applications. In Seq2Seq, many to many RNN architecture is used for decoding. In this, encoder-decoder architecture, input sequence is fed as a vector representation of text to encoder. Then, encoder produces some intermediate representation of information or thought vectors. Subsequently, the thought vector produced by encoder is fed to the decoder as input. Finally, decoder processes the thought vector and converts the sequence one by one word and produces multiple output from the decoder in form of target sequence.

Though, vanilla RNN is default in Seq2Seq and works well for many NLP problems yet, due to higher complexity of language modelling problem, vanilla recurrent neural network cells often fails where long sequence of information needs to be retained, as this information frequently becomes large for larger datasets which creates information bottleneck for the RNN network.

Therefore, researchers use variations of recurrent neural network to handle such problem. Long-Short-Term-Memory (LSTM) is a cell type of Recurrent Neural Network which has empirically shown to work well for language modelling. LSTM has forget gates along with input gates and output gates. This helps remember more relevant and contextual information and discards the rest of the sequence which is desirable in language modelling where dependency within sequence is sparse. Also, instead of using unidirectional cells, bidirectional LSTM cells can perform much better [4].

The encoder-decoder can be diversified into two models, one is retrieval based and another is generative based. Retrieval is much easier because developers use repositories with predefined responses and generative based generates new responses from scratch, they are typical based on machine translation techniques [13].

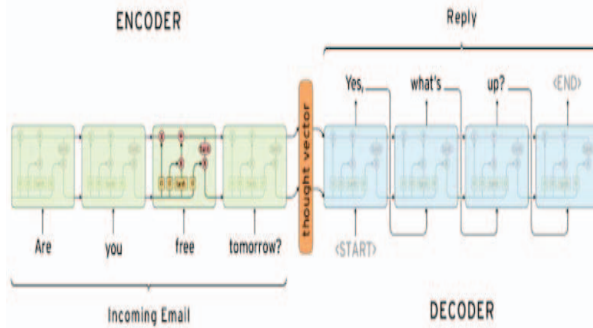


Fig 6: Seq2Seq Model

Another technique, **Neural Attention Mechanism** embedded in Seq2Seq module has significantly improved performance in dialogue generation system and other NLP tasks and thus become industry standard practice. In Neural attention mechanism, each hidden target compares with source hidden state, generates attention vector by calculating score and preserves the attention vector in memory to choose over other candidate. Seq2Seq have also been applied for other NLP tasks including machine translation, text summarization and question-answering and image captioning [4].

H. Google's Neural Machine Translation

Google's Neural Machine Translation (GNMT) model is a module for neural machine translation from and to other language and English. GNMT has also been used for dialogue generation experimentally [12]. It is based on Seq2Seq model which is popular in dialogue generation. GNMT also has many techniques embedded inside its module which are important for intelligent chatbot development. The GNMT model includes, Sequence to Sequence modelling with encoder-decoder architecture built using uni or bidirectional LSTM cells. They also have option for Neural Attention Mechanism, Beam Search, and vocabulary generation using Google's sub-word module. Also, they have option for adjusting the hyperparameters for better model training [4].

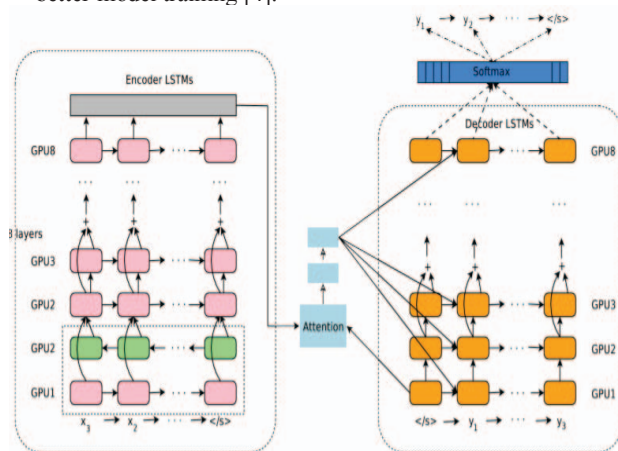


Fig 7: The model architecture of GNMT

The above diagram is a great example of GNMT based work, the left part represents an encoder network and the right part depicts the decoder network and in-between lies

the attention module. The lower layers in the encoder network are bidirectional in nature: the pink nodes collect data from left to right and the green nodes collect data from right to left. There consist residual layers as well which begin from third layer from the below in the diagrammatic representation. There are also certain layers which are unidirectional in nature in the encoder network. For speeding the training process, this model gets divided into many GPUs. In this way each replica is divided into eight different parts and placed separately on eight specific GPUs which belongs to single host machine. For training these bidirectional encoder layers, first these layers compute in parallel. Once the bidirectional models are trained, the program can start with training the unidirectional layers on individual GPUs. Developers must use bottom decoder layers only for gaining recurrent attention context which is later on transmitted to all the other decoder layers, this process helps in storing high amounts of parallelism while running the decoder layers. The softmax layer can also be divided and positioned on different GPUs. Depending upon the size of output vocabulary, developers can either run them on same GPUs just like the encoder/decoder networks or run them on certain specifically dedicated GPUs [9].

I. Deep Reinforcement Learning

Deep Reinforcement Learning is used for Goal-Oriented (GO) Chatbots. These chatbots are used for finding solutions to specific query like making reservations, booking tickets. A GO chatbot has two approaches, one is by using supervised learning with an encoder-decoder that maps user dialogs to responses and another technique is by using reinforcement learning which trains the chatbot through trial and error method with a real user or a rule-based user simulator. The dialog system of GO chatbot consists of three major parts as shown in the image below which are Dialog Manager (DM), Natural Language Generator (NLG) and Natural Language Understanding (NLU). Requests from NLU are given to DM which is further divided into Dialog State Tracker & RL agent. The Backend Database helps in adding useful information whenever its needed and helps in retrieving information time-to-time. Such methods aim to provide Robustness, Flexibility & Reproducibility [17] [18].

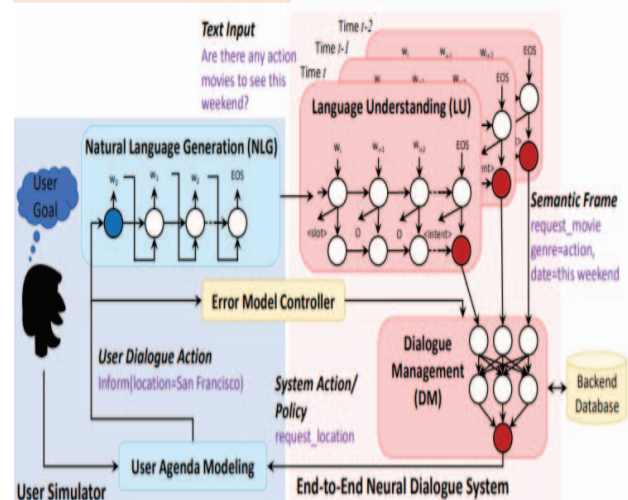


Fig 8: Workflow for Deep Reinforcement Learning

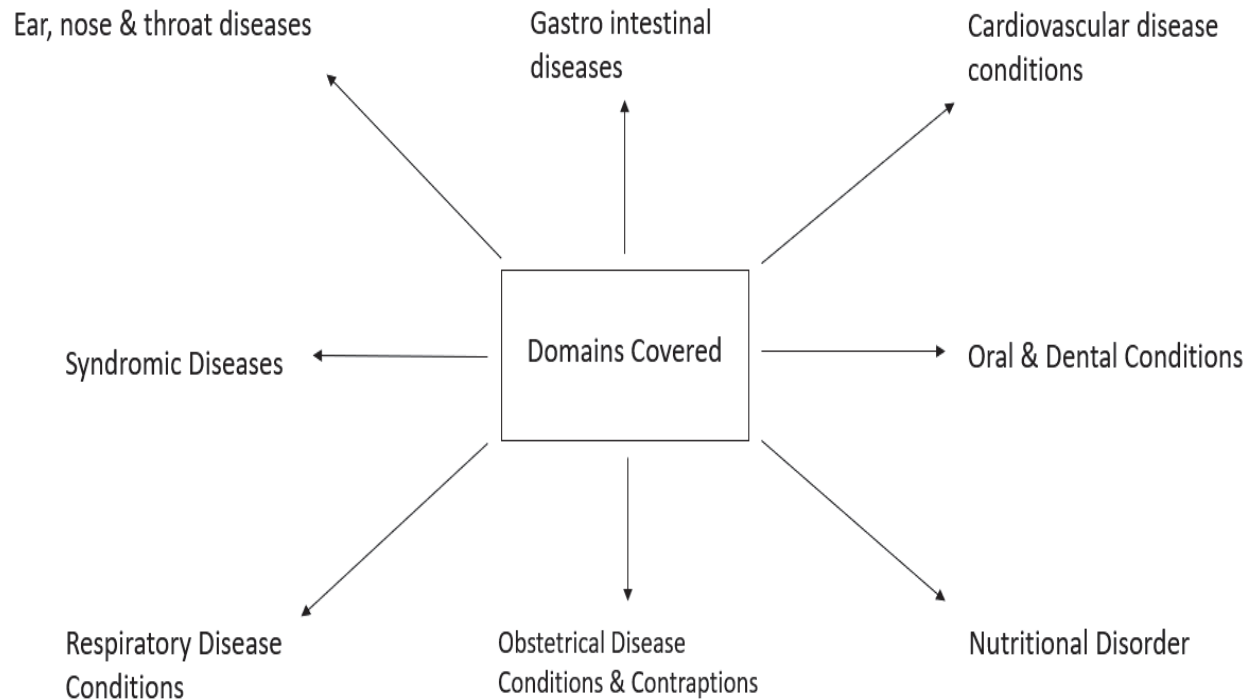


Fig 9: List of medical illnesses covered in bot

We tried to cover as many domains of diseases as we can for our chatbot. This Chatbot aims to diagnose basic healthcare difficulties we usually face in our daily lives following domains are covered in this chatbot, this chatbot gives basic predictive diagnosis which can help patients to get an idea about what disease they might be facing. Such tools will help healthcare industry in future to satisfy the basic customer needs and as technology advances, we hope to see better solutions in future [23].

V. IMPORTANT PACKAGES & PYTHON VERSION

A. Tensorflow

TensorFlow is a software library used for numerical computation by implementing data-flow graphs. Although it was developed by Google Brain Team which is a part of Google's Machine Intelligence research organization for machine learning and deep neural networks research, TensorFlow is still applicable in wide variety of domains. For installing this we have used “pip install tensorflow” command. Some of the advantages of using TensorFlow are Library Management, Debugging, Scalability, Pipelining. It also has a unique technique which allows the program to monitor the training progress of models and tracking several metrics [24].

B. TFLearn

TFLearn is a transparent and modular deep learning library which is built on top of TensorFlow. It has been designed specially to provide a higher-level API to TensorFlow for facilitating and speeding-up experiments, while being fully transparent and compatible with it. Key features of TFLearn include fast prototyping via highly modular inbuilt neural network layers, optimizers, regularizes, understanding high-

level APIs for implementing deep neural networks & metrics also it has full transparency over TensorFlow as all functionalities are built on top of tensors and can be based independently of TFLearn. To install this package, type “pip install tflearn” in Command Prompt [25].

C. NumPy

NumPy library is used in adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. For this work we have worked with a specific version of NumPy which is 1.16.4 because the latest version has compatibility issues with latest version of TensorFlow when we import these libraries. To install type “pip install numpy==1.16.4” in Command Prompt [26].

D. NLTK

Natural Language Toolkit or simply NLTK is one of the most powerful NLP libraries, this library contains packages helps machines in figuring out human language and reply to it with an appropriate response [32]. Character count, Tokenization, Punctuation, Lemmatization, Stemming, Word count are some of its features. NLTK is built to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. To install this type “pip install nltk” in Command Prompt [27].

E. Python 3.6

Python 3.6 is the ideal version we have used for our work as many packages are compatible with this version and currently any higher version of python doesn't support

certain packages which were necessary for this works implementation. Maybe in future packages will support higher versions of python.

VI. PROPOSED METHODOLOGY

A. JSON file for creating intents and training data

The JSON file used in this work has many instances which will help in training the model for our purpose. In this file there will be intents which will be denoted by tags, and within these tags there will be patterns which will be the kind of input user will type, even if the the patterns don't match word by word, the trained model will automatically figure out the exact intent and give a responses which will be nearest to that pattern (highest probability). We have also used "context_set" which will help in grouping similar kind of intents like lets say bot is building conversation with user which is related to his/her disease, so program will group all the symptoms tags related to that disease which will help in finding the exact disease the patient needs to be diagnosed with [5].

B. Importing Libraries in main.py

Important packages have already been discussed above like Numpy, NLTK, TFLearn & Tensorflow. Following are another list of packages necessary for this work :

- Random:

The random module helps in giving access to functions that support many tasks. This random package helps us in giving random responses each time we run the code, these responses are related to the same tag to which we have added to trained pattern [1].

- JSON:

JSON also known as JavaScript Object Notation is a lightweight data-interchange format. Major benefit of using JSON is that it's easy to understand and implement. JSON data can be easily created by machines that is completely language independent. JSON uses similar programming conventions like the C-family of languages which includes C++, C, C#, Java, Perl, JavaScript, and Python. Such properties of JSON make it an ideal choice for data-interchangeable programming language. Importing this package will help in reading and extracting important intents for the training data [30].

- Pickle:

Pickling is used for serializing and de-serializing a Python object structure. Pickling helps in storing the pickled data which can be used again in future. Pickling procedure can convert any python object like list, dict, etc. into a character stream [31].

- Lancaster Stemmer:

The below diagram gives proper representation on how stemming works, any word which comes inside the system gets reduced to its original form or stem word. The reason for doing stemming is because it reduces the burden on processes and makes the program to work faster and give better responses. Sometimes the stemmed words might not hold the exact same morphological meaning, but it can help in reducing its form. Lancaster Stemmer is one of the most

aggressive stemming algorithms because sometimes it can stem words to such an extent that they get translated into strange stems.

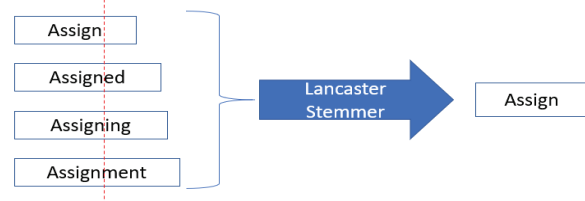


Fig 10: Lancaster Stemmer diagramatic representation

C. Program Algorihtm:

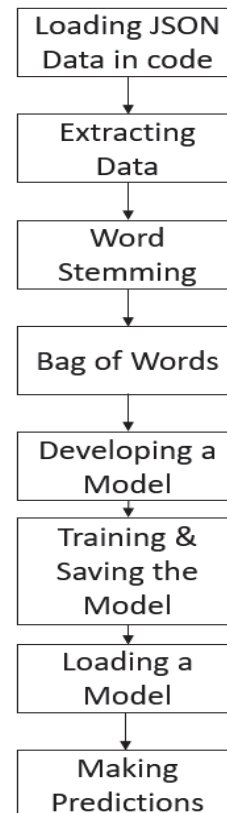


Fig 11: Methodology Implementation

After importing all the necessary data from the JSON file, there is another step called pickling which will store this data so that there is no need to train the model whenever we run the program in future. Then there will be tokenization which means that it will chop the given sentence into smaller parts (token). And store this data in arrays, this program will use two variables one to store the tags and another to store the appropriate responses. Then, the code will loop through our JSON data and extract the data necessary for implementation. For each pattern we have turned it into a list of words using nltk.word_tokenizer, rather than having them in strings. Then there will be implementation of stemming which is an attempt to find the root meaning from a given words for example, the stem word for "cooling" is "cool". This process will help in reducing the vocabulary for our model so that it can train better and try to find more general meaning behind the sentence.

After loading the data and creating a stemmed vocabulary, the next step will be to use the concepts of “Bag of Words” which is a part of Natural Language Processing. Currently, machine learning algorithms and neural networks require numerical inputs. So, our list of strings won’t cut it. Bag of words will help our sentences to represent in numbers. What we are going to do is represent each sentence with a list the length of the amount of words in our model’s vocabulary. Every position in the list will likely represent a word from the vocabulary. If the position in the list stores 1 then that will mean that the word exists in our sentence but if it stores 0 then the word is absent. This is called bag of words because the order in which the words appear is lost in the sentence, we only know the presence of words in our model’s vocabulary.

As well as formatting our input we needed to format our output to make sense to the neural network. Similarly, to a bag of words we had create output lists which are the length of the amount of labels/tags we have in our dataset. Each location in the list will represent distinct label/tag, e.g. 1 can be any of those positions will help in identifying corresponding label/tag if present.

After having pre-processed all of our data we were ready to start creating and training a model. For this chatbot we have used standard feed-forward neural network consisting of two hidden layers. The trained network of dataset will look at a bag of words and return a class which it belongs too (one of our tags from the JSON file). Next step was to define the architecture of the model.

Network building and training model will be the core of our project work. More neurons & hidden layers can be added which can help in optimizing larger training models. The output layers use softmax functions which helps in returning the neuron with highest probability which helps in deciding the accurate response. After doing this, save the

training model so that whenever the bot functions there won’t be any need to retrain it.

While using the model ideally, we want to generate a response to any sentence the user types in. To do this bag of words input is necessary and not string input. Also, our model does not spit out sentences, it generates a list of probabilities for all of our classes. Following were the steps conducted:

- Obtaining user inputs
- Translating or converting it into bag of words
- Making prediction for the model
- Finding class with needed probabilities
- Picking response at random from the class

The bag_of_words function will transform our string input into a bag of words using our created words list. The chat function will be looking into handling prediction from the model and selecting an appropriate response from JSON file.

D. Setting Threshold for eliminating irrelevant questions/words user inputs

As everyone knows that chatbot aren’t trained always to answer any question the user asks. So, there should be an appropriate response whenever the user types a question to which the bot doesn’t have an answer. We are doing this so that the chatbot don’t want to return any inappropriate response which is closest to any tag but unrelated to user’s question. Hence, we will set the threshold probability to 70%. If the responses are above this value, then the response with the highest probability will be returned or else chatbot will return “Sorry I didn’t get it can you please try again”.

VII. RESULTS

User Input	Bot Response
Hi	Hello!
What is your name?	My name is Healthcare Chatbot.
so what do you do?	I would be pleased to assist you on basic healthcare related diagnosis
can you please help me	Hey! no need to panic I am always here for you can you please elaborate all the symptoms you are facing
I am having cough, shortness of breath fever chills muscle pain sore throat lossing taste and smell	It might be possible that you have come in contact with coronaviurs and you seriously need to get a Covid-19 testing. If you have mild symptoms and are otherwise healthy, self-isolate and contact your medical provider or a COVID-19 information line for advice.
Thank you	Any time!
dnkcniwcbi	I don’t get it can you please try again. Thank You!
Bye!!!	Good Bye, I hope I was helpful
quit	(ends the program)

Fig 12: Result table

The model we have built is working fine, adding more intents will help in assessing the user in much better manner. Our chatbot is trained with multiple responses so that whenever someone reuses the bot, they get new responses and not just the old responses which will be accurate to the user's query. Also, multiple user inputs have been coded in the program to help with appropriate responses.

VIII. CHALLENGES & LIMITATIONS OF BOTS

Chatbots are still in the early phases of development and they are heavily relying on artificial intelligence and machine learning so there are still some limitations or challenges they face in recent years [3].

A. Complex Interface

Chatbots are complex programs which are filled with many intents and sometimes few intents match a particular query pattern, this can result in confusion in the system. Also, poor processing can cause time consumption and leave any people annoyed. As more and more people start migrating on such platforms, businesses need to ensure that they have sufficient database, processing power and well build architecture which can withhold many users at one instance [28].

B. Time-Consuming Scenarios

The main purpose of chatbots is to give quick replies to any user queries but sometimes it can take time to fetch the right response to the query if that query doesn't consist the right keywords or the system is not trained to answer that particular query or if its handling many users then the database can get busy and program gets confused. This can lead to bad user experience and make users to switch to other means [28].

C. High Installation Cost

Chatbots save a lot of man-power as they have the ability to interact with many users at any time instant but building and installing such a tool needs a lot of attention and the highly skilled people to ensure that autonomous activities are managed safely, securely and efficiently. As the demand for businesses keep changing and increasing, so will the demand for making better virtual assistant platforms will increase, this can lead to redesigning user interface platforms and backend changes [28].

D. Less decision-making skills

Making right decisions is the most important aspect of any business and when businesses are conducted using autonomous techniques it should be an upmost priority to make the right choices. Chatbots are also infamously known for making wrong choices and businesses should target for making better designs for the architectures of their chatbots [28].

E. Weak Memory Storage and processing

For certain conditions, it is necessary for chatbots to store previous conversational history for handling tasks in better manner. But when system is not designed properly then it can result in users making repetitive conversations

with the bots which can be time consuming and bad for user-experience. Storing previous chat can help in enhancing user experience and suggest better options to users in future [28].

IX. FUTURE SCOPE

Virtual Assistants like chatbots are already making a lot of contribution in the market of digital world. According to a research [8], there are nearly 5.19 billion unique smartphone users, 4.54 billion internet users via smartphones and 3.80 billion users actively using social media. On an average every internet user spends around six to seven hours online on daily basis. With this rising number of consumers on online platform servers a great platform for businesses to connect with their customers. Many restaurants, online ticket booking platforms, small businesses and financial firms are using platforms like Facebook messenger, Slack or WhatsApp business for increasing revenue and providing better customer experience. As businesses keep getting in this market, the demand for bots will keep increasing. Big enterprises and start-ups are all working on improving the frontiers they have achieved in the domain of virtual assistants.

The goal of scientists and engineers currently focuses on making the conversations more helpful and engaging because many times it happens that a bot gives unrelated response and the end user quits using the bot to actually reach out him/herself. Microsoft visions to create a personal assistant which can predict the user's needs, they want to build intelligence that augments human abilities and experiences, it's not going to be about man vs the machines but man with machines. With user's permission, Cortana will be able to read email and make proactive suggestions liking rescheduling a meeting or finding reservation for dinner just by scanning the conversations in user's inbox [7]. This brings us to the point of user privacy and data security; every big enterprise aims to ensure that the user data they get is only used for enhancing the user experience and not for any unethical activity. Modern day chatbots can be improved by analyzing the challenges they face and can be used in various fields like education, healthcare, businesses and many more. [1] Currently implementing technologies like NLU and Deep Learning are still in their early stages to be implemented on a larger scale so to solve this many companies use Conversational Interface which is a hybrid UI where the content is visualized in a chatbot format and users can interact with the system using text, voice or any other natural language interface combined with UI elements like buttons, menus, images, videos, etc.

X. CONCLUSION

Working on chatbots has given us a brief exposure in the domain of Virtual Assistantship and how it will shape our lives in future. After looking at current work and development enterprises and other organizations are making in this field it is definite that chatbots are really going to be an important part of large enterprises as well as small businesses and other organizations. In this paper we discussed about various packages necessary to install, workflow of the code, making data in the intents file, training our model & get a relevant output. We also talked

about the various industry applications of chatbots, relevant work, the common challenges and limitations of it. As technology is growing in various domains, healthcare is one such important domain. And such services will help the patient to get a better understanding of the problems he/she is facing. It will also help in reducing the burden from hospitals when there are many patients to handle. Building a well-trained bot for any services should be the topmost priority for any organizations.

XI. REFERENCES

- [1] M. Dhaiya, "A tool of conversation," International Journal of Computer Sciences and Engineering, Volume 5, Issue 5, 2017, pp. 158-161.
- [2] T. Ruscica, "Chatbot using Deep Learning," Tech with Tim, May 2019, <https://techwithtim.net/tutorials/ai-chatbot/part-1/>.
- [3] "Chatbot," Wikipedia, Wikimedia Foundation, 22 June 2020, <https://en.wikipedia.org/wiki/Chatbot>.
- [4] A. Tiha, "Intelligent chatbot using Deep Learning" Master Thesis: University of Memphis, 10.13140/RG.2.2.14006.75841, April 2018.
- [5] G. Kassabgi "Contextual Chatbots with TensorFlow," Chatbot Magazine, Medium, 7 May 2017, <https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077>.
- [6] M. Mantha, "Conversational AI: Design & Build a Contextual AI Assistant" Towards Data Science, Medium, March 2019, <https://towardsdatascience.com/conversational-ai-design-build-a-contextual-ai-assistant-61c73780d10>.
- [7] C. A. Sottile, "Microsoft Envisions a Future of Bots, AI That Predict Your Every Need", NBC News, March 2016, <https://www.nbcnews.com/tech/tech-news/microsoft-envisions-future-bots-ai-predict-your-every-need-n548151>.
- [8] S. Kemp, "DIGITAL 2020: 3.8 BILLION PEOPLE USE SOCIAL MEDIA", We are Social USA, January 2020, <https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media#>.
- [9] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Yoshikiyo Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", arXiv:1609.08144v2 [cs.CL], Oct 2016.
- [10] S. Dangelo, "Introduction to Watson Assistant," IBM Developers, August 2019, <https://developer.ibm.com/technologies/artificial-intelligence/articles/introduction-watson-assistant>.
- [11] "RASA Architecture," RASA User Guide, RASA Technologies, 2020, <https://rasa.com/docs/rasa/user-guide/architecture/>
- [12] "Google Neural Machine Translation," Wikipedia, Wikimedia Foundation, 30 May 2020, https://en.wikipedia.org/wiki/Google_Neural_Machine_Translation.
- [13] D. Britz, "Deep Learning for Chatbots, Part 1 – Introduction," WILDML, 6 April 2016, <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/>.
- [14] "Introduction to the Messenger Platform," Facebook for developers, 2020, <https://developers.facebook.com/docs/messenger-platform/introduction>.
- [15] "Facebook Messenger," Wikipedia, Wikimedia Foundation, 29 June 2020, https://en.wikipedia.org/wiki/Facebook_Messenger.
- [16] D. Mamgain, "Dialogflow vs Lex vs Watson vs Wit vs Azure Bot | Which Chatbot Service Platform To Use?," Kommunicate, 10 May 2019, <https://www.kommunicate.io/blog/dialogflow-vs-lex-vs-watson-vs-wit-vs-azure-bot/>.
- [17] X. Li, Y. N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "End-to-End Task-Completion Neural Dialogue Systems," Microsoft Research, Redmond, WA, USA, National Taiwan University, Taipei, Taiwan arXiv:1703.01008v4 [cs.CL], February 2018.
- [18] M. Brenner, "Training a Goal-Oriented Chatbot with Deep Reinforcement Learning," Towards Data Science, Medium, 1 December 2018, <https://towardsdatascience.com/training-a-goal-oriented-chatbot-with-deep-reinforcement-learning-part-i-introduction-and-dce3af21d383>.
- [19] "How bots work," Azure Bot Service, Microsoft, 31 January 2020, <https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-basics?view=azure-bot-service-4.0&tabs=csharp>.
- [20] "What Is Amazon Lex?," AWS Developers Guide, Amazon AWS, 2020, <https://docs.aws.amazon.com/lex/latest/dg/what-is.html>.
- [21] "Gartner Says 25 Percent of Customer Service Operations Will Use Virtual Customer Assistants by 2020," Gartner News Room, Press Release, Tokyo, Japan, 19 February 2018, <https://www.gartner.com/en/newsroom/press-releases/2018-02-19-gartner-says-25-percent-of-customer-service-operations-will-use-virtual-customer-assistants-by-2020>.
- [22] "Dialogflow basics," Dialogflow Documentation, Google Cloud 25 June 2020, <https://cloud.google.com/dialogflow/docs/basics>.
- [23] "STANDARD TREATMENT GUIDELINES AND ESSENTIAL MEDICINES LIST," The United Republic of Tanzania, Ministry of Health and Social Welfare fourth edition, May 2013.
- [24] "Tensorflow," Wikipedia, Wikimedia Foundation, 1 July 2020, <https://en.wikipedia.org/wiki/TensorFlow>.
- [25] "TFlearn: Deep learning library featuring a higher-level API for TensorFlow," TFLearn, GitHub, 18 June 2017, <https://github.com/tflearn/tflearn>.
- [26] "NumPy v1.16 Manual," SciPy.org, NumPy, 18 February 2020, <https://numpy.org/doc/1.16/>.
- [27] "Natural Language Toolkit," NLTK 3.5 documentation, 2020, <https://www.nltk.org/>.
- [28] M. Gupta "CHATBOTS: BOON OR BANE?," Bluelupin, January 2018, <https://blog.bluelupin.com/chatbot-advantages-and-disadvantages/>.
- [29] D. Crockford "How Javascript works," Introducing JSON, <https://www.json.org/json-en.html>.
- [30] "JSON," Karina's Web Apps Blog, Wordpress, 20 July 2008, <https://kbrown132.wordpress.com/2008/07/20/json/>.
- [31] S. Sam "Python Pickling", Tutorials Point, 20 Feb 2019, <https://www.tutorialspoint.com/python-pickling#:~:text=Python%20pickle%20module%20is%20used,serialization%20or%20flattening%20or%20marshalling>.
- [32] "NLTK (Natural Language Toolkit) Tutorial in Python", Guru99, <https://www.guru99.com/nltk-tutorial.html>.