**CHATBOT using Deep Neural Network**

…………………………………………0

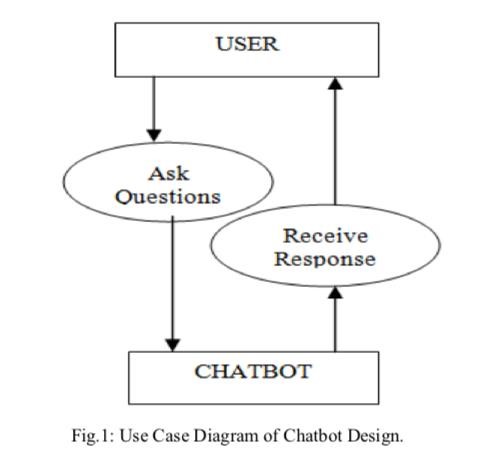
***Abstract*—A Chatbot is artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, and mobile apps or through the telephone. These Chatbots use deep learning models. They use natural language processing models which use deep learning and then find at the bottom here we've got the sequence to sequence models which we'll be interested in at the end.**

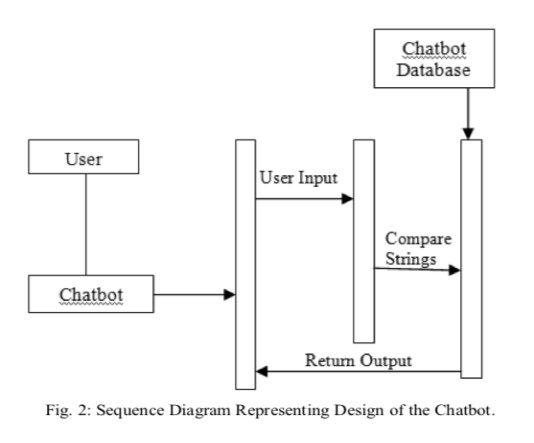
# I. INTRODUCTION

A Chatbot is a service, which is mostly powered by artificial intelligence in which we interact with each other via a chat interface. The service has many applications ranging from functional to fun, and it is already present in any major chat products like Facebook, Messenger, Telegram.

Generally speaking a Chatbot is any software that performs an automated task; however we are interested in the class of bots that live online in chat platforms or on social media called Chatbots.

There are many possible definition of what a Chatbot is and it is perfectly fine because of so many use cases. One of the recent definitions is that it is a Chatbot which can have a conversation with human. For example a user could ask the Chatbot a question or give it an instruction and it could respond or perform an action as appropriate.





## A. Misconception

This definition however often leads to two potential misconceptions.

The first misconception is it has the ability to converse with a human like any other humans but this is wrong because it is simply not possible to achieve that using the current technology. It has been showcased in a lot of sci-fi films but it is still far from reality. This also leads to sky high expectations and it leads to disappointment when these expectations are not met

The second misconception is that a Chatbot communicates using only text or voice. This is not true because there are many instances of various Chatbots that not only use voice and text but also uses graphics to communicate with the user. A large technological company like Facebook, Tencent and Alibaba uses graphical interfaces to interact with their customers.

## B. Difference with Applications

There are a lot of differences between chatbot as an user interface or used as an app. A Chatbot however can be differentiated from an app in the way that the interactions with the bot take place, more or less sequentially (as a conversation), and the bot is used inside a chat app.

A good metaphor for a big difference is human agent and its customers. A Chatbot is different from an app as it has an

“identity” that is actually separate from its interaction with the user. Likewise the agents exist independently when there is no engagement with the customer. This metaphor can be explained with the fact that a single Chatbot could interact with the customer over several different communication channels.

In short a Chatbot is another way of humans interacting with software. Although there are a lot of similarities between apps, websites and Chatbot and they do share most of the functionality But the interaction is very much different when done through a Chatbot.

Nowadays messaging platforms are becoming universally accessible through mobile apps or app portals. Businesses want to find ways to deliver their messages and services in the place that the consumers are which is on chat platforms. Chatbots are the Perfect platform to do this.

## C. Seq2Seq

RNN Encoder-Decoder is the main go to model for Machine translation and dialogue systems which has introduced Sequence to Sequence model in learning phase representations. It consists of two RNNs: An Encoder and a Decoder. The encoder takes a sequence say, sentence as input and processes one symbol say, word at each time step. It encodes only the important information in sequence by converting a sequence of symbols into a fixed size vector while losing the unnecessary information. Each hidden state influences the next hidden state and the final hidden state can be seen as the summary of the sequence. This state is called the context or thought vector, as it represents the intention of the sequence. From the context, the decoder generates another sequence, one symbol (word) at a time. Here, at each time step, the decoder is influenced by the context and the previously generated symbols.When we are using this model, there are a lot of challenges. One of the most challenging part is it cannot handle different variable length sequences. Another challenging part is the vocabulary size. The decoder has to run softmax over a large vocabulary of say 20,000 words, for each word in the output. Even though the hardware is capable, but it usually slows down the training process. Representation of words is of great importance. But how do we represent the words in the sequence? Use of onehot vectors means we need to deal with large sparse vectors due to large vocabulary and there is no semantic meaning to words encoded into one-hot vectors. Let’s look into how we can face these challenges, one by one.

**…**

h

0

h

1

h

2

h

n

g

0

g

2

g

1

**…**

h

0

h

1

h

2

h

n

g

0

g

2

g

1

Hello Ankit Just…. EOS Yes I am back

ENCODER DECODER

## D. RNN

We want to take an example of any sequential data, which can be the stock market’s data for a particular stock. A number of features are used for a simple machine learning model of an Artificial Neural Network may learn to predict the stock prices based on a number of features: the volume of the stock, the opening value etc. While the price of the stock depends on these features, it is also largely dependent on the stock values in the previous days. In fact for a trader, these values in the previous days (or the trend) is one major deciding factor for predictions.

That is when fitting the model for a particular day, there is no consideration for the stock prices on the previous days. In the conventional feed-forward neural networks, all test cases are considered to be independent. This dependency on time is achieved via Recurrent Neural Networks. A typical RNN looks like:

This may be intimidating at first sight, but once unfolded, it looks a lot simpler:

Now it is easier for us to visualize how these networks are considering the trend of stock prices, before predicting the stock prices for today. Here every prediction at time t (h\_t) is dependent on all previous predictions and the information learned from them.

Sequence handling can be done by RNNs to a large extent but not entirely. To be able to build a story and remember it, we need our models to be able to understand and remember the context behind the sequences, just like a human brain. This cannot be done by RNN because RNNs are great when it comes to short contexts.

## E. Limitation of RNN

As we have seen Recurrent Neural Networks work just fine when we are dealing with short-term dependencies.

Most of the times RNNs fail to understand the context behind an input. When we are making prediction in the present Something that was said long before, cannot be recalled .Let’s understand this as an example: Here, we can understand that since the writer has worked in India for 2 years, it is very likely that he may possess a good command over Hindi. But, to make a proper prediction, the RNN needs to remember this context. What happens is the main information may get diluted because of the irrelevant data. This is where a Recurrent Neural Network does not work

Vanishing Gradient is the reason behind this kind of problem. An existing knowledge of how a feed forward neural network learns needs to be looked intoin order to understand this. According to our knowledge for a conventional feed-forward neural network, the weight updating that is applied on a particular layer is a multiple of the learning rate, the error term from the previous layer and the input to that layer. So it results in a product of all previous layer errors which gives the error term for a particular layer. There are many activation functions but in this case we took example of Sigmoid function the small values of its derivatives which occurs in the error function gets multiplied multiple times as we move towards the starting layers. Due to this result, it becomes very difficult to train these layers as the gradient almost disappears as we move towards the starting layers.

A similar case is observed in Recurrent Neural Networks. RNN remembers things for just small durations of time, i.e. if we need the information after a small time it may be again produced, but once a lot of words are feed into it, this information gets lost somewhere. This issue can be resolved by applying a slightly tweaked version of RNNs – the Long Short-Term Memory Networks.

## F. LSTM

One of the things which we do is that most often than not we arrange our calendar for the day, and also we try to prioritize our appointments. Suppose urgency comes we need to make some time and space for it so we mostly know which meetings we can cancel to make time for that possible meeting.

But we find out that a RNN doesn’t do so. If we have to order new information or add onto the existing ones, we apply a function that transforms the existing information completely. Due to this, the entire information gets modified i. e. there is no consideration for what important information is and what not so important information is.

LSTMs on the other hand, make small modifications to the information by multiplications and additions. In LSTMs, there is a mechanism through which all the information flows which is known as cell states. This way, LSTMs can selectively remember or forget things. There are three different dependencies for information at a particular cell state.

We’ll visualize this with an example. Let’s take the example of predicting stock prices for a particular stock. The stock price of today will depend upon:

1. The trend in which the stock has been trading in the previous days, maybe a downtrend or an uptrend.
2. The price of the stock also depends on the previous day, because there are many traders who compare the stock’s previous day price before thinking of buying it.
3. There can be a number of factors that can affect the price of the stock in the present day. This can be a new company policy that is being criticized widely, or a drop in the company’s profit, or maybe an unexpected change in the senior leadership of the company.

These dependencies can be generalized to any problem as:

1. The previous cell state
2. The previous hidden state
3. The input at the current time step

Industries use these dependencies to move products around for different processes and LSTMs use this mechanism to move information.

When it moves through different layers we can do some addition, modification or removal of information, just like the same way product may be molded, painted or packed .

The following diagram explains the close relationship of LSTMs and conveyor belts.

Although this diagram is not even close to the actual architecture of an LSTM, it solves our purpose for now.

We do not manipulate the entire information but rather modify them slowly and slightly, they can forgetand rememberthings selectively due to this property of Lstms

## G. Architecture of LSTMs

By understanding the functioning of a news channel’s team covering a murder story we can visualize the functioning of LSTM. Whenever a new event occurs we take any of the three steps.

Let’s say, we are trying to assume that the murder was done by giving poison to the victim, but the autopsy report that just came in said that the cause of death was due to an impact on the head. We immediately try to forgetthe previous cause of death and all stories that were revolving around this fact.

What can happen if an entirely new suspect is introduced into the picture. A person who had problems with the victim could be the potential murderer. We try to inputthis information.

After a certain time interval, we need to summarize this information which cannot be served by mainstream media these broken pieces of information. So, to outputthe relevant things.

Now let’s get into the details of the architecture of LSTM network:

. A typical LSTM network is comprised of different memory blocks called cells the rectangles that which we are trying to see in the image**.**  There are two states that are being transferred to the next cell; the cell state and thehidden state. The memory blocks are responsible for remembering things and manipulations to this memory is done through three major mechanisms, called gates**.**

## H. Applications

 First one is speech recognition.

So I've got to put them in order of importance in order how many and how often and impactful it is for a lot of lives. These are world changing applications. So speech transcription is a huge one. Whenever we talk into wire phone and try it out as text that is specious script ion so taking in audio form and putting it into text. But not only does it stop there. Remember previously on the Get excited story we talked about some of the chat boards that are or that are like Siri or town or Alexa that are voice based.

Well that same technology goes in there to understand the wave forms and some of the sound we can apply a deep natural language processing and then Reiji put in that second part which is the chat board in order to understand what to do with that text what to do with those instructions.

So speech transcription as it can that is it's already very important is going to become more and more and more important in the years to come.

DEEP

LEARNING

Neural Network

Based Model

Neural

Network

for NLP

Retrieval

Based

NN

Generative

NN

RNN

SEQ2SEQ

CNN

Multilayer

Perceptron

LSTM

# II. PROPOSED WORK

We will build a Chatbot by implementing a state of the art deep NLP model which will be the Seq2Seq and we will implement that with one of the best API which is tensorflow. So this implementation will be done in five steps which are the common five steps when implementing a deeper application or an AI.

We follow a step by step approach.

* Step 1: Getting the Dataset  Step 2: Data Preprocessing
* Step 3: Building the Seq2Seq model
* Step 4: Training the Seq2Seq model
* Step 5: Testing the Seq2Seq model

1. Getting the Dataset

.

We will get the data set which is the Cornell movie corpus. It is a data set of more than 600 movies containing thousands of conversations between lots of characters. And we want to train our model. The model can also be trained on other datasets for some other purposes like for example we will be able to train the same chatbot on a more specific dataset like a calendar assistant or a navigation assistant. There are some more specific applications but this is not what we want do.

We will try a general chatbot to talk about everyday conversations and that’s why movies are perfect because in movies we have a lot of

random conversations general conversations between friends.

1. Data Preprocessing

Data processing is inevitable whenever we build an AI or whenever we build a machinery model we have to make the data set compatible with the model we're going to build. We're going to build a neural network based model and therefore the data will have to have a special format especially for the inputs. Besides we'll have to clean the text because the less we clean it and simplify it the more difficult it will be for a model to train itself to talk like a human. We want it do it the most efficiently so that we can get to step 3, 4 and 5.

1. Building the Seq2Seq model

We will be building the Seq2Seq model which is a state of the art deep NLP model. So we will build it. We will actually build a brain composed of an encoder and then a decoder and we will assemble all of them to build the final brain which has not been trained yet.

1. Training the Seq2Seq model

We will train the model. We will set up a last function to get the optimizer and then apply some to get a grade in the center to update the weight of the neurons of the brain so that it improves its ability to talk with us.

1. Testing the Seq2Seq model

We finally test the model to know once we executed have an interface where we can ask some questions and then the chatbot will answer and we just test the Chatbot by observing its answers and see how is capable of conversing with us.

**Data Preprocessing**

**Getting the Dataset**

**Building the Seq2Seq model**

**Train**

**ing the Seq2Seq model**

**Testing**

**the Seq2Seq model**

III. RELATED WORK

A chatbot implemented in TensorFlow based on the new sequence to sequence (NMT) model, with certain rules seamlessly integrated.The core of ChatLearner (Papaya) was built on the NMT model, which has been adapted to fit the needs of a chatbot.

Highlights and Specialties:

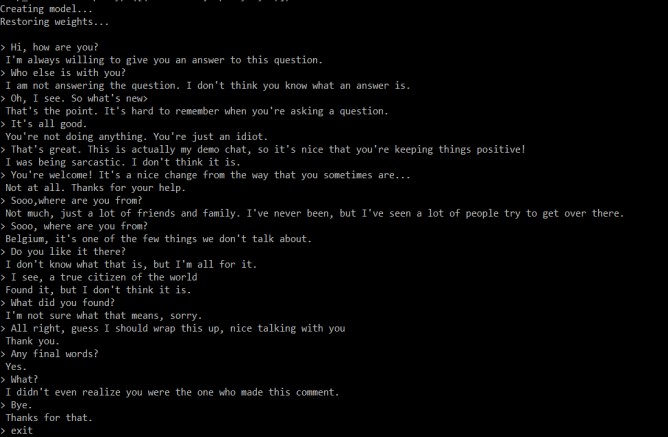
1. The Papaya Data Set for training the chatbot. We can easily find tons of training data online, but we cannot find any with such high quality.
2. The concise code style and clear implementation of

the new seq2seq model based on dynamic RNN (a.k.a. the new NMT model). It is customized for chatbots and much easier to understand compared with the official tutorial.

1. The idea of using seamlessly integrated

ChatSession to handle basic conversational context.

# Papaya Conversational Data Set



Papaya Data Set is the be

st (cleanest and well

-

organized)

free English conversational data we can find on the web for

training a chatbot. Here are some details:

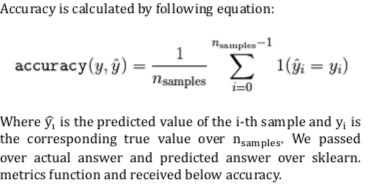
1.

The data are composed of two sets: the first set was handcrafted, and we created the samples in order to maintain a consistent role of the chatbot, who can therefore be trained to be polite, patient, humorous, philosophical, and aware that he is a robot, but pretend to be a 9-year old boy named Papaya; the second set was cleaned from some online resources, including the scenario conversations designed for training robots, the Cornell movie dialogs.

2. The training data set is split into three categories: two subsets will be augmented/repeated during the training, with different levels or times, while the third will not. The augmented subsets are to train the model with rules to follow, and some knowledge and common senses, while the third subset is just to help to train the language model.

# Results

While per-response accuracy calculates the percentage of correct responses, per-dialog accuracy, where a dialog is considered to be correct if and only if every response within it is correct, counts the percentage of correct dialogs.



Training Accuracy: 0.866678031242 Validation Accuracy:

0.856304621164

Below screen shots shows examples of predictions of our model for test examples.

# CONCLUSION

A chatbot is one of the simple ways to transport data from a computer without having to think for proper keywords to look up in a search or browse several web pages to collect information; users can easily type their query in natural language and retrieve information. In this paper, information about the design, implementation of the chatbot has been presented. From the survey above, it can be said that the development and improvement of chatbot design grow at an unpredictable rate due to variety of methods and approaches used to design a chatbot. Chatbot is a great tool for quick interaction with the user. They help us by providing entertainment, saving time and answering the questions that are hard to find. The Chatbot must be simple and conversational. Since there are many designs and approaches for creating a chatbot, it can be at odds with commercial considerations. Researchers need to interact and must agree on a common approach for designing a Chatbot. In this project, we looked into how Chatbots are developed and the applications of Chatbots in various fields. In addition comparison has been made with other Chatbots. General purpose Chatbot must be simple, user friendly, must be

1. [https://medium.com/@BhashkarKunal/conversatio nal-ai-chatbot-using-deep-learning-how-bidirectional-lstm-machine-reading-38dc5cf5a5a3](https://medium.com/@BhashkarKunal/conversational-ai-chatbot-using-deep-learning-how-bi-directional-lstm-machine-reading-38dc5cf5a5a3)
2. [https://medium.com/datathings/the-magic-of-lstmneural-networks-6775e8b540cd](https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd)
3. [https://colah.github.io/posts/2015-08-](https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

[Understanding-LSTMs/](https://colah.github.io/posts/2015-08-Understanding-LSTMs/)

1. [https://ieeexplore.ieee.org/abstract/document/8289 883](https://ieeexplore.ieee.org/abstract/document/8289883)

1. [https://link.springer.com/chapter/10.1007/978-1-](https://link.springer.com/chapter/10.1007/978-1-84628-666-7_14)

[84628-666-7\_14](https://link.springer.com/chapter/10.1007/978-1-84628-666-7_14)

1. https://www.superdatascience.com/pages/deeplearning-chatbot

easily understood and the knowledge base must be compact. Although some of the commercial products have recently emerged, improvements must be made to find a common approach for designing a Chatbot.

## REFERENCES

1. <https://www.expertsystem.com/chatbot/>
2. [https://www.analyticsvidhya.com/blog/2017/12/fu ndamentals-of-deep-learning-introduction-to-lstm/](https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/)
3. [https://hub.packtpub.com/build-generative-chatbotusing-recurrent-neural-networks-lstm-rnns/](https://hub.packtpub.com/build-generative-chatbot-using-recurrent-neural-networks-lstm-rnns/)
4. <https://www.expertsystem.com/chatbot/>