Chatbot and its Practical Applications in the Materialistic World

Gundapu Nitish Kumar, Devavarapu Sreenivasarao, Shaik Khasim Saheb

Abstract**: *These days Chat has become the new way of conversation and changed the way of life and the view that the world used to see before and due to Industrial revolution 4.0 , the gradual increase in machine learning and artificial intelligence fields has gone to higher and many companies are reaching customers to get their products with more ease . This is where chatbots are used. It all started with one question! can machines think? The concept of chatbots came into existence to check whether the machines could fool users and make them think that they are actually talking to humans and not robots. On the Other hand, with the Successes Rate of Chat bots, Different companies Started using machines for having conversations with their customers about everything which made their work simpler and reduced the need of man power. There are many different types of building a chatbot but this paper will mainly concentrate on building a Chatbot using TensorFlow API in python***

Keywords **: *Chatbot, Industrial Revolution, Machine Learning, Artificial Intelligence, TensorFlow, API, Python.***

I. INTRODUCTION

A Chat Bot is a computer program which helps develop conversations with humans, deploying Natural Language Processing in terms of textual, audible and interactive messages. In today's era, many Chat Bots are seen to be revolutionizing the way machines talk to humans, such as Siri, Alexa and Google's Voice Assistant. In simpler terms, a Chat Bot is just a piece of code which takes an input from the user and tries to match the input with similar statements existing in a database comprising of a multitude of such statements, and delivers an output which relies on the model containing a pattern of such statements, usually referred to as the 'brain' of the Chat Bot. The present times have seen the evolution of various kinds of Chat Bots, depending on their means of usage, and so, this paper explains various facets involved in the building of a Chat Bot which uses a data set to train itself and responds accordingly. This paper hence primarily focuses on the development of a user-friendly Chat Bot practically using Google's Deep Learning Framework, TensorFlow in Python.

A. Introduction to Chatbot

Conversation has turned an essential aspect people's lives, and so, many researchers are working today in order to construct a perfect interactive conversation between humans and machines. However, the implementation of conversations in an AI Dependent Systems is a tedious task, owing to the complexity of its design. In the initial stages of development of Chat Bots, patterns were written in a language known as AIML (Artificial Intelligence Markup Language) [1], in which the patterns are written into a file and adopted by a code to construct a conversation. This is a pattern oriented method, since the pattern taken as input is directly searched for in the AIML File.

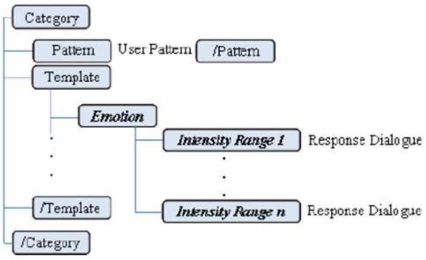


Fig. 1. Composition of an AIML Pattern

An AIML file is a collection of category tags. Each category tag is considered a query and there are numerous category tags depending on the number of queries. These category tags comprise of two main portions – the pattern and the template. The input taken is matched with the pattern, and the template is returned as the answer to the query as shown in Figure 1. The augmentation of Chat Bots by AIML patterns made their modeling simpler. However, they lacked efficiency because of the method of implementation. In most cases the AIML powered Chat Bots were able to answer only those questions to which relevant patterns were mentioned in the AIML File. For instance, if a pattern for the query, "What is your name?" is written in the AIML File, the Chat Bot would only be able to answer that question, but not some other question with the same meaning, such as "Tell me your name." Here is where the concept of bigram calculations [2] came into existence. For the given input, probability calculations of other sentences are made in order to find out similar sentences in the database, i.e.

Manuscript published on November 30, 2019.
* Correspondence Author

Gundapu Nitish Kumar*, is currently pursuing B.Tech Degree program in Computer Science & Engineering in Sreenidhi Institute of Science and Technology, Affiliated to Jawaharlal Nehru Technical University Hyderabad, Telangana, India, PH-7032580182.

Devavarapu Sreenivasarao is currently working an Assistant Professor in Computer Science & Engineering Department in Sreenidhi Institute of Science and Technology and his area research includes Medical Image Processing, Machine Learning. PH-9866014581.

Shaik Khasim Saheb is currently working as Assistant Professor in Computer Science & Engineering Department in Sreenidhi Institute of Science and Technology, and his area research includes Medical Image Processing, Machine Learning. PH-9642097865.

the response returned is of that sentence which has the highest probability. The probability of all the sentences is computed using the below equation.

$$\rho(\varphi) = \prod_{i=0}^{n} \rho(\varphi_i|\varphi_0 \ldots\ldots \varphi_{i-1})$$

However, in the present times, Chat Bots are extensively being used in industries and companies to enhance communication with their customers and improve the accessibility of their products. A multitude of websites on the internet are shifting to Chat Bots from the conventional, archaic techniques of customer care, just like industries which rely on machines to get the task done swiftly and efficiently. Nevertheless, the conventional techniques of creating a Chat Bot fail to work in such instances, owing to the complexity of writing patterns for millions of queries, which is certainly a herculean task. Hence, the only solution to this issue is to build a model which takes an input and returns an output by training itself on datasets and learns to construct replies by itself. This means, we require a deep learning model which learns by itself instead of being explicitly programmed to reply. The following sections elaborate the construction of a Chat Bot, relying on deep learning frameworks. Chat Bots are mainly of two types

1. Generative based (from a Dataset).
2. Retrieval Based (from a Database).

A Retrieval Chatbot need to have a large amount of data in order to train itself and respond to the user's input, hence it requires database in order to retrieve data. but a Generative Based Chat Bot trains itself on a dataset to give the response to the user. This paper will show how to create a chat bot of generative type using seq2seq model which works on recurrent neural networks. but both types of chat bots require substantial amount of resources to get trained and executed. Even after the training process is done, the outputs are not perfect. Hence the inputs are normalized to get the accurate output.

B. Introduction to TensorFlow

TensorFlow is a free open source library designed in order to manage the work flow in different tasks. It was created by Google’s Brain team for its internal usage. It is made open source in the year 2015. TensorFlow is also used in differentiable programming which is widely used in AI. TensorFlow was created initially to control the data flow due to its extensibility over large data. TensorFlow uses a computational graph for executing its tasks. For any TensorFlow model, its computational graph is made in order to execute the model. This paper will show the execution of seq2seq model to build a chatbot which is available in TensorFlow library.

II. METHODOLOGY

The following steps [3] are adopted in order to build a Generative Chat Bot.

Step 1: Choosing a Data set.
Step 2: Data Preparation and pre-processing.
Step 3: Graph Building.
Step 4: Neural Network.
Step 5: Model Selection.
Step 6: Optimizers.
Step 7: Training the Model.
Step 8: Executing.
Step 9: Testing and Results.

A.Choosing a Data set

This paper demonstrates the usage of the Twitter dataset for the purpose of completing this task. This data set consists of lots of useful Twitter Conversations which can be useful for the analysis and the training of the Chatbot. This data set is not especially designed for training chat bots, but it can be used to make simple conversations between humans and machines.

B. Data Preparation and Pre-processing

Data preparation in the context of building a Chat Bot refers to the conversion of data which is the form of text present in datasets to utterance pairs – question and answer pairs used to train the Chat Bot. As mentioned above, the demonstration shown in this paper will use Python. For the Data preprocessing, analyzing and interpretation, nltk and re modules of Python are made use of. The input data must initially be converted either into lowercase or uppercase, in order to be preprocessed. Firstly, unwanted special characters, white spaces are removed from the input. If the input is still found to contain too many numbers and additional, insignificant words, such words too are removed depending upon the input to get the generalized input form. These generalizations are derived using regular expressions and most optimizations are done using the module nltk.

C. Graph Building

For training a model, it must be defined. In the process of graph building, the hidden layers are defined to give the accurate output. The demonstration shown in this paper uses a predefined model named seq2seq for natural machine translation. Using the Dataset API, the data preparation pipeline can be defined and batched. All that needs to be done is wrapping the data using the Dataset API. Now, some data operations should be done on the dataset like Tokenization, Word to Index, Shuffling, and Batching, which convert the entire dataset into utterance-pairs, thereby simplifying processing.

D. Neural Network

The Neural Network used in this paper is an RNN. A Recurrent Neural Network (RNN) [4] is a type of a neural network that can take a sequence D=(d_1, ...,d_n) as an input and produces a sequence of encoded states L =(L_1, ...,L_n)by using the process of recurrence . It is also called as unrolling as shown in Figure 2 at every stage the front part of the network takes x_i andh_{i-1}as input and returns an encoded state h_i as output. At every stage i, the hidden layer h_i is updated by below Equation.

$$l_i = f(Al_{i-1} + Bd_i)$$

Where A and B are the vectors which contains the weights of the neural network, where f is a nonlinear activation function which represents a hyperbolic tangent function. LSTM’s [6] were designed to improve the performance and to resolve the problems faced by the Vanilla version of RNN. The Vanilla version of RNN is not used because of its problem with Vanishing gradient [5]. As the steps involved in the unrolling process increase it becomes harder for a simple recurrence neural network to learn and to remember information.

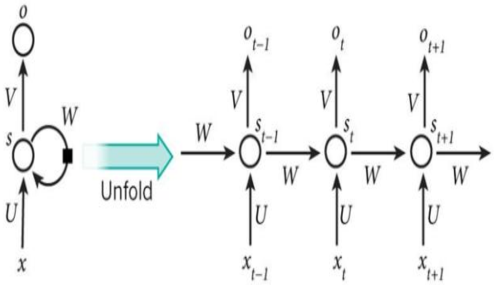


Fig. 2. Unfolding of an RNN over 3 time-steps

In the above mentioned figure, x is the input sequence, o is the output sequence. LSTMs are enabled with gates in order to regulate the dataflow present in the neural network. These gates consist of information related to the operations done on the input and the previous encoded output such as the matrix multiplications, non-linear functions etc. GRUs are mainly used for language modeling, because of its ability of preserving information for different and more than one sentence when the network is being unrolled for every word in the provided input. An important property of an RNN is that the components relates to the activation function remains constant when the network is being unrolled. This makes the model eligible for its usage in Language Modeling. When this model is trained to learn the probability Distribution of vocabulary v it will be well suited for the natural language processing. The probability distribution function that can be used to predict the next word in the sentence is given by below equation,

$$\rho(x_{ij} | x_{i-1} \dots . x_i) = \frac{\exp\,(v_j h_i)}{\sum_{j-1}^{k} \exp\,(v_i h_i)}$$

Hence for each value of j from 1,...,k, v_j represents the rows of the weighted matrix and the sequence (x_{i-1},...x_1) represents the input given to the model and h_i represents the encoded state of the neural network for every value i .so the probability function above results the accurate words for the corresponding input . The training part of the neural network is done using truncated back propagation [8] algorithm. Essentially the error is back propagated for each step. The error can be calculated by using a loss function, which calculates how the calculated values would be when compared with the true labels.

$$Loss(y_j, \overrightarrow{y_j}) = -y_i\, log(y_i)$$

After calculating the loss, the training process is improved by adding optimization techniques such that the neural network gives the output having maximum accuracy. The final outcome from the decoder will be a vector consisting of 1's and 0's, which will be further processed to get the accurate output which resembles required output for the corresponding input. The Seq2seq is a very useful model and these days many of the natural language processing systems uses this model to get the accurate output.

E. Model Selection

The model that will be used is seq2seq. Seq2seq [10] was the first model introduced for the implementation of machine translation using neural netw- orks. Before the model was introduced, the machine translation was done in a different way. Each sentence that was given as input was splited into words and converted to its outcome expected language giving no regards to the respective language's grammatical pattern. Seq2seq has radically changed this process by using concepts of deep learning. This model takes the input splitted words and input neighborhood for the training process while translating to other language. Now a days, this model is used for a many different useful applications like conversational models, image captioning, text summarization etc.

Seq2seq Model Working

The Seq2seq model takes a sentence (The flow of Characters) as an input and generates an output sentence. The process is done by using RNNs (recurrent neural networks). The RNN (Vanilla version) is not used widely because of its problem with vanishing gradient. Instead, more advanced version GRU (or LSTM) is used. The Model is developed with an idea of talking two inputs to get the output where one of them is from the sentence provided by the user and other from the previous given output. The seq2seq model consists of two major RNNs - The Encoder and The Decoder, the working is shown in the figure 3.

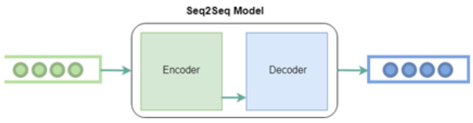


Fig. 3. The Encoder and Decoder Model

Encoder

It is an RNN which uses deep neural networks to convert the given input sequence of words into the sequence of encoded vectors where each hidden state represents the depended factors of the given word.

Decoder

It is an RNN similar to the encoder where it takes the encoded vectors generated by encoder as an input, its own encoded states and present word in order to produce the next encoded vector which is used to predict the next word for the corresponding input. These are the Additional Components of seq2seq model.

Attention

If a hidden state is given to the decoder as a single vector the computation will be easy but the computation becomes difficult if the input is given as the sequence of encoded vectors. Here, the attention mechanism is adopted in order to select the input sequence relatively and selectively.

Beam Search

Always a word which is considered as the output will have the highest probability compared to other output words. but this way of taking output does not always gives best results because of greedy algorithms.

To rectify this, Beam search mechanism is applied which suggests the possible translated output at each and every step. This process is done by making a tree of k-results.

Bucketing

Bucketing is done in order to reduce the memory usage, for example, if the maximum length for the output is assigned as 1000 and the output sentence is just 40 words it leads to the enormous memory usage. To reduce this bucketing is used. We make ordered pairs like (4,6), (7,15) etc which 7 is the input length and 15 will be maximum length of the output that will be generated.

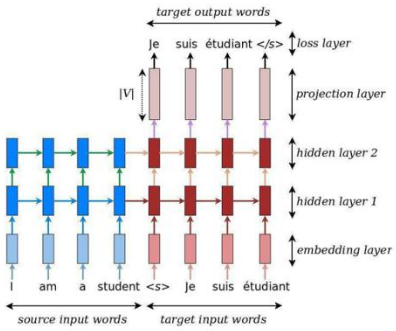


Fig. 4.The Working of the Seq2seq model using RNN

F. Optimizers

Using the loss obtained, gradients can be computed, which train the model using an optimizer. We define an operation known as optimizer which calculates gradients and uses them to train the model. We also add to more components to the optimizer like cosine decay and gradient clipping to operate the training process of the model. Cosine decay is used to analyze the growth rate mechanism and gradient clipping is used to ignore exploding gradients created.

G. Training the Model

Training the model involves initializing all variables and starting to train the model. We have to save the state of the model for every epoch and calculate average loss for each and every epoch to obtain accurate results. This Process may take a lot of time, depending on the hardware configuration of the machine used for the training process.

H. Executing

The Execution part of the Chatbot involves creating a seq2seq model for NLP. But the python's Tensor Flow module consists the model predefined. Hence it made the task easier and more efficient compared to other methods. The number of epochs is 50 and the accuracy is nearly 95 % which turns out to be pretty accurate compared to other models.

I. Testing and Results

Training process of the model took more time than expected but the results were accurate not exact. To test the resulting model different inputs were given, and as expected some of the answers were good and some of them were not that good. This model was made in python and the results were demonstrated using nltk, TensorFlow, NumPy libraries.

III. DISCUSSIONS AND RESULTS

After taking the input from the user, the input is generalized in to such a way that the outcome after the generalizations would be a string with no special characters and exaggerated words. First the input is taken as a string and the white spaces from first and last are removed and converted into lowercase to make the data processing easier.Given table 1 is some of the examples of the processed strings

Table-I: Eliminating white spaces and converting the strings into lower case.

Input	Output
Hi HOw aRe YOu	hi how are you
HeLlOWorlD	hello world
I`M FIne and Good	i`m fine and good
I WIll eat Rice	i will eat rice
ILIkeCriCKet	i like cricket

After the input is converted into lower case, all the Special Characters such as !, @, = etc are replaced with the white spaces in String Which Makes the input to be more readable and easier to process than compared with the sentence with the special Characters. The respective output for the Corresponding inputs is shown in the Table 2.

Table-II: Taking out the Special Characters and replacing them with the white spaces

Input	Output
hi@how(are you	hi how are you
Hello34World	hello world
i`m fine-and good	im fine and good
i=will eat*rice	i will eat rice
i!like)cricket	i like cricket

Then after the punctuations and the grammatical errors in the Sentences are removed using the nltk.stem() function which helps correct sentences and convert them into grammatically accurate ones.

Table-III: After the Correcting the input grammatically

Input	Output
hii how are u	hi how are you
Hello. , world	hello world
i`mm fine and goood	im fine and good
i will eattt rice	i will eat rice
iikecricket	i like cricket

After the input normalization is done, the input is split into a group of words and these words are passed into the seq2seq model. The seq2seq2 model is made of two recurrent neural networks named encoder and decoder. The encoder takes the input and converts the input list of words into an encoded vector using the concept of recurrence as shown in the figure 5.

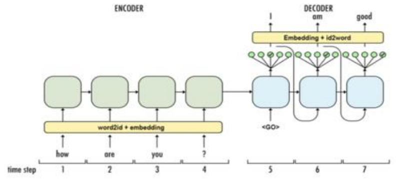


Fig. 5. Sequence of Networks in RNN

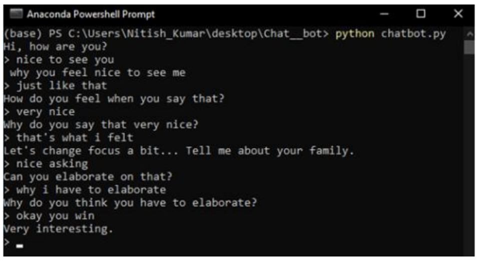
After this, the decoder decodes the encoded vector taking two inputs one from the vector and another from the previous input, which forms a grammatically accurate output from the model.

In the entire process of building the generative Chat Bot, it is observed that the most time-consuming part is the training part, as the time taken for training depends not just on the model, but also on the architecture of the machine. When the Chat Bot is ready to be asked questions, outputs are delivered by it as shown in the table 4.

Table-IV: Outputs after the model is rendered

Input	Output
hi , how are you	I am fine
Do you know me	How can I?
will you marry me	Can you elaborate on that?
What is your Gender	i prefer not to say
bye	have a great time

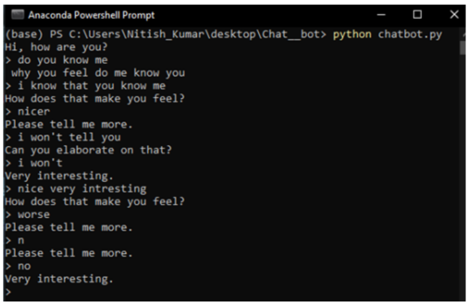
It is observed that reasonably precise outputs are given for the inputs delivered to the Chat Bot. The following figures depict the model demonstrated in this paper. The training part of this model took 50 epochs and in due course of advancement of training, the outputs obtained to the questions asked improve, resulting in nearly accurate outputs.



```
(base) PS C:\Users\Nitish_Kumar\desktop\Chat__bot> python chatbot.py
Hi, how are you?
> nice to see you
 why you feel nice to see me
> just like that
How do you feel when you say that?
> very nice
Why do you say that very nice?
> that's what i felt
Let's change focus a bit... Tell me about your family.
> nice asking
Can you elaborate on that?
> why i have to elaborate
Why do you think you have to elaborate?
> okay you win
Very interesting.
> _
```

Fig. 6. The practical implementation of a Generative chat bot using TensorFlow

When the training process was undergoing for each iteration (epoch), the accuracy was increasing and the loss was decreasing. At First, the results were pretty poor but after some epochs the outputs produced were relevant to the corresponding input given. There are many algorithms and models that can be tried on to create conversational software but, the seq2seq model had given very good results compared to other models.



```
(base) PS C:\Users\Nitish_Kumar\desktop\Chat__bot> python chatbot.py
Hi, how are you?
> do you know me
 why you feel do me know you
> i know that you know me
How does that make you feel?
> nicer
Please tell me more.
> i won't tell you
Can you elaborate on that?
> i won't
Very interesting.
> nice very intresting
How does that make you feel?
> worse
Please tell me more.
> n
Please tell me more.
> no
Very interesting.
>
```

Fig.7. The practical implementation of a Generative chatbot using TensorFlow

IV. CONCLUSION

As demonstrated in this paper, developing Chat Bots using frameworks of deep learning make them powerful, efficient, well-structured and accurate. Though this paper describes the development of a Generative Chat Bot using RNNs, many other techniques can be used to do the same. A point worth noting is that frameworks of Deep Learning have revolutionized the ways in which Chat Bots are built and work, and hence, the world would look up to the making of Chat Bots working on more efficient algorithms, to enhance communication between a human and a machine.

REFERENCES

1. Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research,3(Feb):11371155.
2. Bottou, 2010, Large-scale ma- chine learning with stochastic gradient descent. In proceedings of COMPSTAT2010, pages 177186. Springer.
3. Cho et al., 2014, Cho, K., VanMerrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
4. G.Neubig," NLP Programming Tutorial 2 - Bi- gram Language Models", Presentation Moduleof Nara Institute of Science andTechnology.
5. Hochreiter, S. (1998). The van- ishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge- Based Systems,6(02):107116.
6. Hochreiter,S. AndSchmidhuber, J.(1997). Long short termmemory. Neural computation, 9(8):17351780.
7. Https://omarito.me/building-a-seq2seq-conversational-chat-bot-using-tensorflow/, 2007
8. NanoDano,https://www.devdungeon.com/content/ai-chat-bot-python-aiml
9. Rumelhart, D. E.,Hinton, G. E., Williams, R. J., et al. (1988). Learning rep- resentations by back-propagating errors. Cognitive modelling.
10. Werbos, P. J. (1990).Backpropagation through time: what it does and how to do it. Proceedings of the IEEE,78(10):15501560.

AUTHORS PROFILE

Mr. Gundapu Nitish Kumar is currently pursuing B.Tech Degree program in Computer Science & Engineering in Sreenidhi Institute of Science and Technology, Affiliated to Jawaharlal Nehru Technical University Hyderabad, Telangana, India. His main research work focuses on Machine Learning and Neural Networks

Mr. Devevarapu Sreenivasarao, currently working as an Assistant Professor in the department of CSE in Sreenidhi Institute of Science and Technology since 2014. He did Master of Technology from JNT University Hyderabad, India in year 2012. He is a research scholar in Annamalai University which was located in Chidambaram, Tamilnadu, India He has published more than 15 research papers in various peer reputed international journals..His main research work focuses on Medical Image Processing, Machine Learning.

Mr. Shaik Khasim Saheb currently working as an assistant Professor in the department of CSE in Sreenidhi Institute of Science and Technology since 2014. He did masters from VIT University, Tamilnadu, India. 2014. He is a research scholar in Annamalai University which was located in Chidambaram, Tamilnadu, India. He has published more than 10 research papers in various peer reputed international journals. His main research work focuses on Medical Image Processing, Machine Learning.

Retrieval Number: D8396118419/2019©BEIESP
DOI:10.35940/ijrte.D8396.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication