IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 8, Issue.12, December 2020 E-ISSN: 2347-2693

# Proposal of a Generative type Travel Chatbot using Seq2Seq model

**Subhadeep Jana[1*], Souradeep Ghosh[2]**

[1]Department of Computer Science & Engineering, Government College of Engineering and Ceramic Technology, Kolkata, India
[2]Department of Electrical Engineering, Heritage Institute of Technology, Kolkata, India

[*]*Corresponding Author: jsubhadeep1999@gmail.com, Tel.: +91-8240262186*

DOI: https://doi.org/10.26438/ijcse/v8i12.2126 | Available online at: www.ijcseonline.org

Received: 19/Dec/2020, Accepted: 21/Dec/2020, Published: 31/Dec/2020

***Abstract*—** Artificial Intelligence and Machine Learning can be cited as one of the greatest technological advancements in this century. They are revolutionizing the fields of computing, finance, healthcare, agriculture, space, tourism. Powerful models have achieved excellent performance on a myriad of complex learning tasks. One such product of AI is a chatbot. A chatbot is an intelligent software which can simulate a conversation with a user like a real human being. Chatbots have found their use in customer service, recommender systems, smart appliances, etc. Chatbots can be broadly divided into 2 types: Retrieval and Generative. Retrieval type chatbots are trained to provide the best fit answer from a database of predefined responses, whereas, generative type chatbots can generate the final answer from a training corpus. This paper proposes the design and implementation of a generative type travel chatbot using seq2seq model, which can generate answers to the user queries based on Kolkata tourism.

***Keywords*—** Chatbot, Machine Learning, Neural Network, Deep Learning, NLP

## I. INTRODUCTION

A chatbot is an intelligent software that can simulate the conversation with a user in natural language. It can convincingly emulate the way a real human being would converse with another human being. Chatbots have the ability to learn from their experiences. They will integrate all the experience learning into its skill set and help the user to find its required result with effiency. In the past decade, the use of chatbots has grown manifold. They are being used in handling customer queries, technical consultancy, recommending products in the e-commerce domain, virtual counseling of students, etc.

Chatbots are broadly classified into retrieval and generative type. Retrieval type chatbots have a knowledge base of predefined responses from which it chooses the best answer for the user query. A machine learning model is used to recognize the intent and entities of the user query, and respond with a suitable answer. They are generally closed domain chatbots. A generative type chatbot does not use any predefined repository. Instead, it generates a text response based on the training corpus using natural language generation. They are generally open domain chatbots.

We have designed a generative type travel chatbot which majorly focuses on the Kolkata tourism domain. It has been designed on the seq2seq model, also called the encoder-decoder model, using LSTM (Long Short Term Memory) for text generation from training corpus. We have detailed the architecture, methodology and working of our chatbot in the paper.

## II. RELATED WORK

The concept of chatbot was first introduced by Joseph Weizenbaum in 1966 at MIT. He created the first chatbot in the history of computer science which was named ELIZA [1]. It used pattern matching and substitution methodology to give fixed responses to the users. The creation of ELIZA pioneered the way to further advancements in the field of chatbots. After ELIZA, there were many other successful bots made.

PARRY [2] was constructed by American psychiatrist Kenneth Colby in 1972. The program imitated a patient with schizophrenia and worked via a complicated system of assumptions, attributions and emotional responses triggered by changing weights assigned to verbal inputs. Racter [2] was another interesting chatbot program written in 1983 by William Chamberlain and Thomas Etter. It was labeled as a story-telling chatbot. Racter was the first computer program to write a book. Jabberwacky [3] was created by developer Rollo Carpenter in 1988. It worked on contextual pattern matching and aimed to simulate a natural human conversation in an entertaining way.

A.L.I.C.E (Artificial Linguistic Internet Computer Entity) [4] was created by Richard Wallace in 1995. The chatbot worked with the XML schema known as Artificial Intelligence Markup Language (AIML). It has won the prestigious Loebner Prize three times.

© 2020, IJCSE All Rights Reserved 

Mitsuku is a chatbot created from AIML technology by Steve Worswick in 2005. It is a five times Loebner Prize winner and is often deemed as the most intelligent chatbot ever. It inherits the traits of A.L.I.C.E chatbot.

In 2017, **Minghui Qiu et al** presented a paper entitled "*AliMe: A Sequence-to-Sequence Rerank-based Chatbot Engine"* [5], which presented how a chatbot can be used in the e-commerce domain. The bot is employed in the Alibaba online site and services millions of customer queries a day, mostly in Chinese and English. AliMe integrates a hybrid approach based on Information Retrieval and seq2seq generation model.

*"goTripper Chatbot for Tourism"* by **Monalisha Bandyopadhyay et al** [6] details the architecture of a compact travel chatbot using Apache OpenNLP. The proposed system consists of three basic modules: NLU (Natural Language Understanding) module, state machine module and NLG (Natural Language Generation) module.

**Bai Li et al** presented a paper "*Real-world Conversational AI for Hotel Bookings"* [7] in 2017, which details an AI system to search for and book hotels through text messaging.

Myra, an AI powered chatbot from MakeMyTrip, is a very popular travel chatbot. It gets suggestions and alerts of rail/flight bookings, baggage details, and even suggestions for car bookings.

The Booking Assistant chatbot uses AI technology to help answer customers' questions. It is programmed to answer frequently asked questions on payment, date changes, transportation, pet policies and Internet availability in hotels.

AskDISHA is an AI powered chatbot launched by Indian Railways for its travellers. Users can enquire about booking e-tickets, timetable of trains, reservation status, cancellation and refund.

## III. METHODOLOGY

Natural Language Generation can be achieved by a variety of algorithms and models [8]. Some of them are Markov chain [9], Vanilla RNN, LSTM [10], Transformer [11]. In our proposed chatbot, we have designed a seq2seq model, also known as an encoder-decoder model, using LSTM. Seq2seq models were introduced first in the paper "*Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation"* by **Kyungyun Cho et al** in 2014 [12]. We have used Python programming for the development. The basic architecture of our chatbot is depicted in Fig 1.

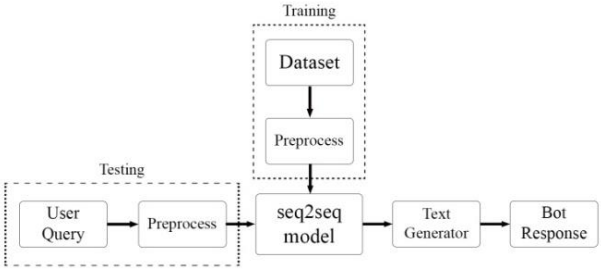


Fig 1 - Basic architecture of our chatbot

The major steps carried out are listed in an order:

### 1. DATASET COLLECTION

Data is the most important ingredient for any neural network. The more data it gets to work on, the better it performs. For our travel chatbot, we curated 1084 pairs of human response – bot response questionnaire on Kolkata tourism, primarily on locations and food. The dataset is spread across 2 text files. A part of the dataset is given in Fig 2.

| *human_responses.txt* | *bot_responses.txt* |
|---|---|
| Hi | Hi there, how are you? |
| Am fine. How are you? | Am fine. |
| What is your name? | My name is Kolly |
| What can you do? | I am a travel chatbot and can tell you about famous locations, food, restaurants in Kolkata. |
| Tell me more about Victorial Memorial. | The Victoria Memorial is a large exquisite marble building in Kolkata which was built between 1906 and 1921, dedicated to the memory of Queen Victoria, then Empress of India. |

*Fig 2 - Dataset*

### 2. PREPROCESSING

We use regular expressions to remove punctuations, except comma, from the sentences. The reason we keep comma is that it helps separate the multiple proper nouns which may appear in bot response, especially for a travel chatbot. Hence we treat comma as a unique token. Then the sentences are converted to lower case to reduce ambiguity. We again use regular expressions to separate comma from their attached words. Finally, a tokenizer is used on the input and output corpus to create the input and output vocabulary set, containing all unique tokens. The dataset after preprocessing looks like in Fig 3.

| *human_responses.txt* | *bot_responses.txt* |
|---|---|
| hi | hi there , how are you |
| am fine how are you | am fine |
| what is your name | my name is kolly |
| what can you do | i am a travel chatbot and can tell you about famous locations , food , restaurants in kolkata |
| tell me more about victorial memorial. | the victoria memorial is a large exquisite marble building in kolkata which was built between 1906 and 1921 , dedicated to the memory of queen victoria , then empress of india |

Fig 3 – Dataset after preprocessing

© 2020, IJCSE All Rights Reserved 

We add a 'START' token at the beginning and 'STOP' token at the end of the bot responses so that our model knows where to start and end text generation.

## 3. VECTORIZATION

Vectorization is a process of converting words into numbers. We declare 4 Python dictionaries – Input Features, Reverse Input Features, Output Features and Reverse Output Features. The feature dictionaries will help us to vectorize the text and the reverse feature dictionaries will help us to build text from a vectorized format. There are various ways of vectorizing a text; we use one-hot vectorization in the process. One-hot vectorization, as the name suggests, makes use of only 0 and 1. Three 3-Dimensional arrays are created to store the vectorized forms of input and output corpus. To give an insight into the vectorization, let us consider the input sentences to be ["Hi, how are you?", "What is your name?"] and output sentences to be ["Am fine", "My name is Kolly"]. Input tokens are [',' , 'are' , 'hi' , 'how' , 'is' , 'name' , 'you' , 'your' , 'what'] and output tokens are ['am' , 'fine' , 'kolly' , 'my' , 'name' , 'is']. They are arranged alphabetically. Our dictionaries would look like in Fig 4:

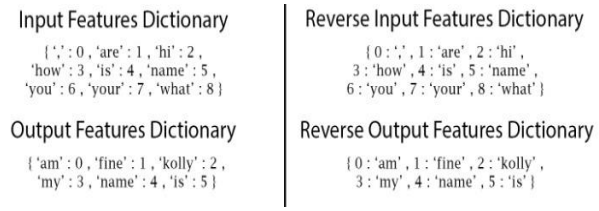


Fig 4 – Vectorization dictionaries example

After vectorization of the input corpus, the 3-D array will look like in Fig 5:

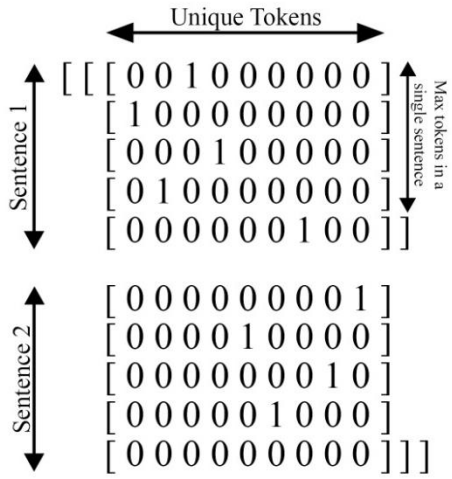


Fig 5 – Vectorized 3D array of sample input corpus

The dimensions of the array are: No of sentences in the corpus x Maximum number of tokens in a sentence x Total number of unique tokens. As you can see, we do not need to perform padding as it is already padded. Padding [13] is basically the process of structuring all the vector sequences to the same length for better fitting. This is generally done by adding extra zeros to the sequences if needed. In Fig 5, the 2nd dimension of the vectorized array is the maximum number of tokens in a sentence. We take the maximum so that every other sentence can be padded to the maximum length. The reason we use 3 arrays is a method called teacher forcing which is used by the Seq2Seq model while training, explained in detail in model training section.

## 4. MODEL CREATION

Seq2Seq model is generally used for text generation from a training corpus [14], text summarization, question answering systems, image captioning, conversational modeling [15, 16], and machine translation applications. It is also called encoder-decoder model because it contains two RNN (Recurrent Neural Network) structures, one for encoding and another for decoding [17, 18]. It takes as input a sequence of words and generates an output sequence of words. Although the vanilla version of RNN is rarely used, nowadays more advanced versions like LSTM (Long Short Term Memory) & GRU (Gated Recurrent Unit) are utilized. The reason for this is vanilla RNN suffers from the problem of vanishing gradient, where it cannot remember long term dependencies. This can be overcome by using a LSTM cell. A basic LSTM cell architecture is given in Fig 6:

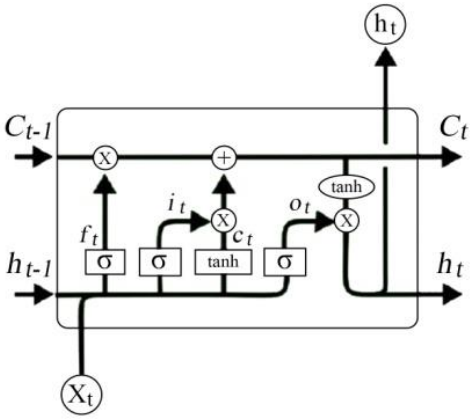


Fig 6 – Basic LSTM cell architecture

There are 2 states of an LSTM cell, cell state and hidden state. The cell state is the memory of the LSTM cell, and hidden state is the output of the cell. In Fig 6, we can see each LSTM cell has 3 inputs: $C_{t-1}$, $h_{t-1}$ and $X_t$. $C_{t-1}$ stands for the cell state input from a memory cell in timestep t-1, $h_{t-1}$ stands for the hidden state input from a memory cell in timestep t-1 and $X_t$ is an input in timestep t. A LSTM cell comprises of 3 gates: input gate (adds new data to the cell), output gate (outputs cell data) and forget gate (erases cell data). In the input gate, we decide to add new content from the present input to our present cell state. In the output gate, we decide what to output from our cell state. In the forget gate, we take decisions about what must be removed from the $h_{t-1}$ state and keep only the relevant content.

We used this LSTM cell to construct our RNN encoder and decoder. We took dimensionality of 512 for our LSTM layer. It defines the dimensions of the output space. The model predicts a word given in the user input and then

© 2020, IJCSE All Rights Reserved 

each of the successive words is predicted using the probability of that word to occur. The two network are trained together to increase the conditional probability of the output sequence given an input sequence. The encoder outputs a final state vector which becomes the initial input for the decoder. We used Tensorflow and Keras to build our Seq2Seq model. Tensorflow is an open source end-to-end machine learning platform and Keras is an open source Python library that provides an interface for modeling artificial neural networks.

The architecture of the model is given in Fig 7:

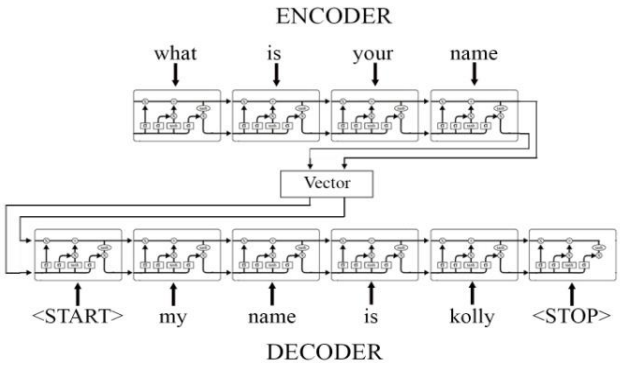


Fig 7 – Seq2Seq model architecture

**5. TRAINING**

As stated previously, we use three 3-Dimensional matrices of one-hot vectors to train our model. We named them Encoder input, Decoder input, Decoder output. We use two matrices for the decoder to implement a method called teacher forcing [19, 20]. It is a common technique used in training Seq2Seq models, or RNN's in general. Teacher forcing is a technique where the output word is passed as the next input to the decoder. As a sequence comprises of words in a particular order, the prediction of a single word depends on the previously predicted words. So one wrong prediction can break the entire sequence and a wrong answer will be shown by the bot. Teacher forcing remedies this. We take help of an input token from the previous timestep to help the model train for the current output token.

We fit the data into our created model using Adam optimizer and Categorical Cross-Entropy loss function. We use Encoder input, Decoder input as feature values and Decoder output as target values. We use 20% of our training data for cross-validation.

**6. TESTING**

The testing of our Seq2Seq model entails receiving a user query, processing it and giving out an answer. The model we created for training implements teacher forcing, where the output sequence is known. But it will not work for an unknown query. Hence we define a testing setup containing a Seq2Seq model in individual pieces. We build an encoder model with encoder inputs and output states. We build a decoder model with decoder inputs, the final states from the encoder and decoder output states. Lastly, we create a function that accepts the user query and generates a response using the model. The function creates the vectorized matrix of the input text and feeds the matrix into the testing encoder model. We retrieve the output states and then we pass them into the decoder, which becomes our initial decoder hidden states. Our word prediction starts and as each word is predicted, we update the hidden state of the decoder so that we can use previously predicted words to help predict the successive ones. The words predicted are initially in a vectorized format which we decode using the Reverse Output Features dictionary. We stop once we hit the <STOP> token as it indicates the end of a sentence.

## IV. RESULTS AND DISCUSSION

Given below is a sample conversation with our Kolly travel chatbot in Fig 8.

```
>>> Hi, I'm Kolly. I can help you with Kolkata tourism. Ask me about historical locations & restaurants. Type bye to exit me. What would you like to know?
>>> hi
>>> hi there , how are you
>>> am fine
>>> am fine
>>> what are you
>>> i am a travel chatbot and can tell you about famous locations , food , restaurants in kolkata
>>> okay
>>> okay , happy to help
>>> do you have a name?
>>> my name is kolly
>>> where can i go in kolkata?
>>> victoria memorial , howrah bridge , indian museum , fort william , birla planetarium , marble palace , st pauls cathedral , science city , birla mandir , eden gardens , jorasanko thakurbari , rabindra sarovar , shobhabajar rajbari , kalighat temple , botanical gardens , alipore zoo , prinsep ghat , eco park , nicco park , park street , south park street cemetery , national library , kumortuli , nandan , mothers wax museum , biswa bangla gate , salt lake stadium
>>> enlighten me about howrah bridge
>>> howrah bridge is one of the busiest cantilever bridge built over the hooghly river , it connects the twin cities of howrah and kolkata
>>> thanks
>>> okay , happy to help
>>> can you suggest me places for chinese food?
>>> okay , happy to help
>>> chinese?
>>> tung nam eating house , tung fong restaurant , mainland china , hong kong chinese restaurant , haka , golden joy restaurant , barbq , china town restaurant , big boss restaurant , golden dragon , beijing restaurant , royal jade , kim ling , krystal chopstick , chowman , chung wah , jimmys kitchen , hatari restaurant , asia kitchen , buddha bites
>>> you are great
>>> i am a travel chatbot and can tell you about famous locations , food , restaurants in kolkata
>>> okay thanks for letting me know
>>> okay , happy to help
>>> what about indian museum?
>>> the indian museum is the ninth oldest museum in the world , oldest museum in india and the largest museum in india it has rare collections of antiques , armors and mughal paintings
>>> nice
>>> okay , happy to help
>>> okay bye
>>> okay , have a great day
```

Fig 8 – Sample conversation with our chatbot

© 2020, IJCSE All Rights Reserved 

As we can see the chatbot is able to understand the intent of most of the human queries and answer them accordingly. The Seq2Seq model efficiently builds up the response by predicting the correct tokens. When asked "where can i go in kolkata?" it replies with all the tourist destinations we have mentioned in our dataset. When asked "enlighten me about howrah bridge" it replies with the info on Howrah Bridge. The question "can you suggest me places for chinese food?" could not be answered correctly, contrary to the question "chinese?" which was answered correctly. It may be due to lack of dataset or training, as both questions mean the same. It replies correctly to exclamations and exit cues. Hence we can see the chatbot could recommend places to travel to, give info about the places, and also suggest restaurants for a particular cuisine, limited to our dataset. It makes use of the training data, learns efficiently and generates responses.

We tried to fit the model using six optimizers, a loss versus epochs graph of which is given below in Fig 9:

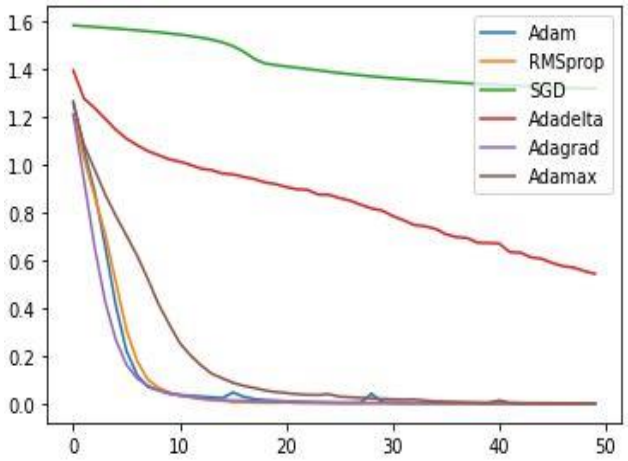


Fig 9 – Loss vs. Epochs graph for optimizers

From the graph we can notice that the SGD (Stochastic Gradient Descent) and Adadelta optimizer could not converge within 50 epochs. The other four optimizers converged to their global minima's, performing fairly similar to each other. Using Adam optimizer and Categorical Cross-Entropy loss function, we got an accuracy of around 53% on training. The hyper-parameters had the following values: default learning rate of 0.001 for the optimizer, batch size equal to 10, epochs equal to 50. We performed some tuning and fixed on these values as they worked the best. For 1084 training examples, the accuracy we got was decent.

It could not answer some questions correctly which is due to lack of training dataset. The training data needs to be robust and well defined, especially for a travel chatbot. A generative chatbot is as good as its training corpus. If we increase the amount and variety of data, the accuracy would surely improve. Also we ran the training for 50 epochs, which maybe increased for better fitting.

## V. FUTURE SCOPE

Travel chatbots, apart from their intelligent conversation feature, have many other features too. With that being said, our travel chatbot has multiple areas of development and future work. Our chatbot currently has a limited knowledge base of a few locations, cuisines and restaurants of Kolkata. It can be expanded to more locations, tourist spots, heritage sites, shopping destinations, restaurants and eateries, navigation, hotel information - everything a user would expect out of a travel chatbot acting as a virtual guide. More human response bot response pair can be added to the database to make the chatbot a complete one, giving it the power to handle a variety of queries. Elaborate exception handling can be integrated to manage the out of context questions that the user may ask. The only punctuation we used was a comma to reduce ambiguity. Further work can be done to support all punctuations.

For user query input, we can integrate voice input besides text input. We will use a speech input module to process and parse the voice input before sending it to the testing model, and a text to speech generator for obtaining the audio of the text generated response. For navigation, we can integrate the Google Maps API into our chatbot, which can help us to gather data like geo-location, latitudes, and longitudes of the user from the Maps database. It can help in navigating the user to a specific area of Kolkata or suggest routes to reach a specific site. We can integrate the MediaWiki's API into our chatbot through which it can access Wikipedia information on a topic and relay that to the user. A hotel booking API can be added to facilitate the process of searching and booking hotels, availability of rooms, hotel services, and automated payment process.

## VI. CONCLUSION

In conclusion, we propose a generative type travel chatbot system which utilizes a Seq2Seq model to generate valid responses from a training corpus and answer user queries. We could replicate a human conversation well using a Seq2Seq model with LSTM, as presented in the paper. This study can be a base for creating more advanced conversational travel chatbots, working on a much more elaborate and robust dataset. Our chatbot is limited only to Kolkata, but we can design a chatbot for any travel location. The chatbot can help new tourists to explore, and act as a virtual guide. The chatbot will be easy to operate, user-friendly, and will help a user to get a concise clear answer to his travel related query. Generative type chatbots are an epitome of what an intelligent system looks like. It learns from its experiences and knowledge; and tourism is one such domain where the use of generative type chatbots will grow in the coming years.

## REFERENCES

[1] J. Weizenbaum, "*ELIZA - A Computer Program for the study of Natural Language Communication between Man and Machine*", Communications of the ACM, Vol.**9**, Issue.**12**, pp.**36-45**, **1966.**

© 2020, IJCSE All Rights Reserved 

[2] T. Zemcik, "*A Brief History of Chatbots*", In the Proceedings of International Conference on Artificial Intelligence, Control and Automation Engineering (AICAE 2019), **China**, pp.**14-18**, **2019.**

[3] A. Mittal, A. Agarwal, A. Chouksey, R. Sriwas, S. Agrawal, "*A Comparative Study of Chatbots and Humans*", International Journal of Advanced Research in Computer and Communication Engineering, Vol.**5**, Issue.**3**, pp.**1055-1057**, **2016.**

[4] B. AbuShawar, E. Atwell, "*ALICE Chatbot: Trials and Outputs*", Computacion y Sistemas, Vol.**19**, Issue.**4**, pp.**625-632**, **2015.**

[5] M. Qiu, F. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang, W. Chu, "*AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine*", In the Proceedings of the 55th Annual Meeting of the Association of Computational Linguistics, **Canada**, pp.**498-503**, **2017.**

[6] M. Bandyopadhyay, M. Sahoo, M.L. Rangani, J.K. Mirji, "*goTripper Chatbot for Tourism*", International Journal of Computer Sciences and Engineering, Vol.**7**, Special Issue.**14**, pp.**36-40**, **2019.**

[7] B. Li, N. Jiang, J. Sham, H. Shi, H. Fazal, "*Real-world Conversational AI for Hotel Bookings*", In the Proceedings of 2nd International Conference on Artificial Intelligence for Industries (AI4I 2019), **USA**, pp.**58-62**, **2019.**

[8] S. Santhanam, S. Shaik, "*A Survey of Natural Language Generation Techniques with a focus on Dialogue Systems – Past, Present and Future Directions*", CoRR, Vol.**abs/1906.00500**, **2019.**

[9] F.S. Al-Anzi, D.M. AbuZeina, "*A Survey of Markov Chain models in Linguistics Applications*", In the Proceedings of the 5th International Conference on Advanced Information Technologies and Applications (ICAITA 2016), **UAE**, pp.**53-62**, **2016.**

[10] A. Sherstinsky, "*Fundamentals of Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) Network*", CoRR, Vol.**abs/1808.03314**, **2018.**

[11] D. Varshney, A. Ekbal, G.P. Nagaraja, M. Tiwari, A.A.M. Gopinath, P. Bhattacharya, "*Natural Language Generation using Transformer in an Open Domain Setting*", In the Proceedings of the 25th International Conference on Natural Language & Information Systems, **Germany**, pp.**82-93**, **2020**.

[12] K. Cho, B.V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "*Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*", In the Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), **Qatar**, pp.**1724-1734**, **2014**.

[13] M. Dwarampudi, N.V. SubbaReddy, "*Effects of padding on LSTMs and CNNs*", CoRR, Vol.**abs/1903.07288**, **2019.**

[14] P. Mazare, S. Humeau, M. Raison, A. Bordes, "*Training Millions of Personalized Dialogue Agents*", In the Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), **Belgium**, pp.**2775-2779**, **2018**.

[15] P. Jayachandran, K. Nawas, C. Jackson, S. Ramanath, R. Prabhakaran, "*Towards building a Neural Conversation Chatbot through Seq2Seq model*", International Journal of Scientific and Technology Research, Vol. **9**, Issue.**3**, pp.**1219-1222**, **2020**.

[16] G. Gnanaguru, "*Programming a Chatbot in Python using Emotional Cognitive Conversational Agent Architecture (ECCAA)*", International Journal of Computer Sciences and Engineering, Vol.**7**, Issue.**3**, pp.**510-516**, **2019.**

[17] Z.C. Lipton, J. Berkowitz, C. Elkan, "*A critical review of Recurrent Neural Networks for sequence learning*", CoRR, Vol.**abs/1506.00019**, **2015.**

[18] I. Sutskever, O. Vinyals, Q.V. Le, "*Sequence to Sequence Learning with Neural Networks*", In the Proceedings of 28th International Conference on Neural Information Processing Systems (NeurIPS 2014), **Canada**, pp.**3104-3112**, **2014**.

[19] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, "*A Scheduled Sampling for Sequence prediction with Recurrent Neural Networks*", In the Proceedings of 29th International Conference on Neural Information Processing Systems (NeurIPS 2015), **Canada,** pp.**1171-1179**, **2015**.

[20] A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio, "*Professor Forcing: A New Algorithm for training Recurrent Networks*", In the Proceedings of 30th International Conference on Neural Information Processing Systems (NeurIPS 2016), **Spain,** pp.**4608-4616**, **2016**.

## AUTHORS PROFILE

Mr. Subhadeep Jana is currently pursuing his Bachelor of Technology degree in Computer Science and Engineering from Government College of Engineering and Ceramic Technology, Kolkata, India. His research interests lie in Machine Learning, Natural Language Processing.

Mr. Souradeep Ghosh is currently pursuing his Bachelor of Technology degree in Electrical Engineering from Heritage Institute of Technology, Kolkata, India. His research interests lie in Machine Learning, Natural Language Processing.

© 2020, IJCSE All Rights Reserved 