International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8, Issue-1C, May 2019

CAISY: Chatbot using Artificial Intelligence and Sequential Model with YAML Dataset

Manoj Kumar M V, Prajwal J M, Shyamanth Kashyap, Rahul G, Pavan R Nargund

Abstract: ***This paper presents a method for constructing a software chat robot named "CAISY". It trains itself to answer queries for any domain it has been trained to handle. CAISY uses word embedding, Sequence 2 Sequence model, and Long Short Term Memory neural networks. CAISY is capable of responding to test queries under a maximum of 5 millisecond delay. The exciting feature of Caisy is it could be applied to any domain where the data is available for training the model. For the demonstration, this paper presents the CAISY trained for Normal Conversation. CAISY has been optimized by minimizing the loss value with the help of variation of hyperparameters.***

Index Terms**: *Word embedding, Sequence 2 Sequence, Long Short Term Memory(LSTM), Hyperparameters***

I. INTRODUCTION

A chat robot (Chatbots) [1] is a computer program which converses effectively with humans. They act as a conversation partner for humans and also would behave like one. Chatbot are typically used for various purposes like customer service or for information acquisition on various topics. Artificial intelligence [2] technologies are disrupting customer care and engagement processes. Chatbots are essential to comprehend changes in customer care services provided and in many usual queries that are most frequently enquired

Some Chatbots use well advanced natural language processing, but most of the simpler chatbots map input with the keywords on the dataset and then effectively post a reply with the most matching keyword set, or the most similar pattern it identifies from the database. CAISY presented in this paper acts on question/query input by the user, then process it accordingly to find the optimal answer with respect to background knowledge. From the industrial revolution to mobile telephony and the internet revolution, India has traditionally been a late adopter of advanced technologies. However, this statement does not prove the context of chatbot construction. India has produced highly sophisticated chatbot software since the inception of chatbot development. It is evident that a few of chatbots developed in India are well-positioned to compete, not just in the country, but across global markets.

Revised Manuscript Received on May 20, 2019.
Manoj Kumar MV, Prajwal JM, Shyamanth1Kashyap, Rahul G, Pavan R Nargund
Department of Information Science and Engineering,Nitte Meenakshi Institute of Technology, Yelahanka, Bengaluru - 560064

Deployment of a chatbot plays a vital role by strengthening the customer service by delivering real value and fast, effective support which in turn strengthens your business. All companies require automation of customer care tools that lead to an increase in productivity and as a result, improves the relationship with their customers. To achieve all the aforementioned goals, we aim to develop a chatbot named CAISY to address the following,

- Elucidate the techniques to capture word embedding.
- Construct a sequence to sequence model to build CAISY.
- To understand how loss varies over varying hyper parameters.

The upcoming sections of this paper are organized as follows, Section 2 briefs the literature related to the chatbot, Section 3 gives the overview of the framework of CAISY, Section 4 describes experimental setup and implementation, Section 5 describes results and analysis of results. Finally, Section 6 gives a brief conclusion of the paper.

II. EVOLUTION OF TECHNOLOGY TO DESIGN CHATBOT

It all began in the year 1950 when Alan Turing published an article entitled "Computer Machinery and Intelligence" [6]. This article led to numerous questions like, "Can machines show human-like behavior?", "Can they think like humans?" and many more.

This resulted in the development of human-like conversing robots referred to as Chatbots or Chatterbots. In the year 1966, ELIZA [7] a Natural Language Processing computer program designed at MIT AI Lab by Joseph Weizenbaum. It was the first bot to be created which had a more human-like conversation. In the year 1972, ELIZA met another bot named PARRY [8] in an ICCC event. PARRY was made to counterfeit a person with paranoid schizophrenia based on concepts, conceptualizations (accept, reject, neutral) and beliefs.

Between the 80s - 90s numerous chatbots were developed - some predominant ones are ALICE, Dr. Sbaitso [9]. Later in the start of the 21st century, i.e., in the year 2001 Smarter Child [10], an intelligent bot was developed. The chatbot was launched on different platforms like AOL, IM and MSN Messenger which could carry out fun conversations

Retrieval Number: A10210581C19/19©BEIESP
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

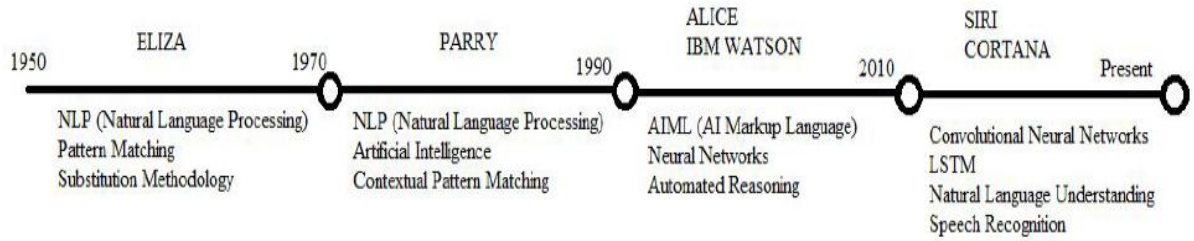


Figure 1: Evolution of technology for designing the chatbot

In 2006 Watson [11] was developed by IBM. Watson tokenizes the questions into different keywords and fragments sentences to find analytically related phrases. Apple, Microsoft, and Amazon came up with a bot named Siri [12], Cortana [13] and Alexa[14] respectively. These assistants use voice recognized input as questions and a natural language processing user interface is built to answer those questions or to make recommendations or to perform actions as obtained from the internet source.

As presented in the paper "Long Short-Term Memory" (LSTM) [15] LSTM is designed to overcome the back-flow error problems. LSTM's can learn to bridge the time intervals which are more than a thousand steps which includes many cases like noisy and input sequences which cannot be compressed. The major advantage is that this can be attained without the loss of any short time or lag capabilities.

In the paper titled "Efficient Estimation of Word Representations in Vector Space" by Mikolov et al. and his colleagues tells that there are numerous techniques to represent words as vectors. Continuous Bag of Words (CBOW) model and skip gram model are the two main models used for word embedding. CBOW helps in predicting the central word given the surrounding word as input. Skip gram model helps in predicting the next words given the previous words. From the paper, we can infer that skip gram model is suitable for the design of the Chatbot. We can see that the accuracy and execution speed of skip gram model is much more efficient than that of CBOW or any other model. Hence in our project, we have incorporated skip gram model for word embedding which suits as an efficient way for getting the required results.

Having the good idea of literature, we limit the scope of this paper strictly to the deployment of the LSTM model for training CAISY, and using same to answering queries. Usage of an attention model for handing queries is not in the scope of this paper.

III. FRAMEWORK

Overview of CAISY steps are illustrated in the fig. 2. CAISY involve following steps, for preprocessing the data set and optimally predicting the answer.

- **Collection of Dataset**: We collect the dataset including a different set of topics and make it into a file with YAML format.
- **Regularize**: We regularize the dataset by removing - and -- present in the YML format file. We divide the sentences starting with -- as queries and sentences starting with - like responses.
- **Tokenization:** It is the process of dividing the sentences into words and we send those words for word embedding.
- **Word Embedding**: Word Embedding is the process of transformation of words into vectors.
- **Generation of Batches:** The vectors obtained from the previous step are processed in batches and are fed into the encoder-decoder model for learning.
- **Encoder-Decoder model**: In this model, we input the words to the encoder and obtain the next predicted word through the decoder.

CAISY is Trained in the various domain through the datasets and is able to predict the right kind of output or response to the query asked.

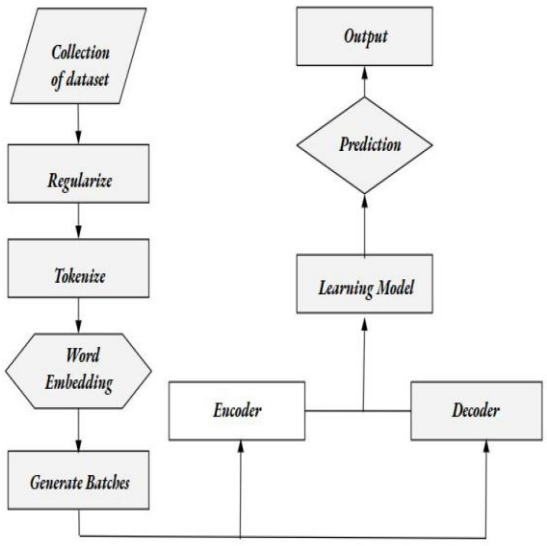


Figure 2: Framework of CAISY

Retrieval Number: A10210581C19/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV. IMPLEMENTATION

A. *Data Acquisition:*

Data is one of the most important concepts in building a chatbot. The type of data defines the application of the chatbot. To help the need of our area of application for the chatbot we have used YAML format dataset which contains information on normal conversations, general topics and other data obtained through Wikipedia.

YML is a format of a file which contains normally asked user queries starting with - (hyphen) and the response to that question starting with - - (double hyphen).

B. Word Embedding

Word Embedding is a technique used in predictive NLP model. Word Embedding helps in transforming vector representation of words into a dense vector space by which we can identify the similarities between the words as shown the figure 3.

We have split the large sentences into words and have removed special characters along with punctuation marks. After tokenization of words, we have embedded the words into vectors using python enumerate function which assigns numbers to words in an orderly fashion.

All the query dataset will be having an even integer embedding and responses with odd integers. By the help of this embedding, it is possible to predict the words which should appear in a sentence by using the keyword through an attention mechanism.

C. Generate Batch

In this process, we have padded the input words with 0 to match that of the output words. We have also embedded the words without response as an unknown keyword. We generate two sets of arguments with input training words and output training words which are divided into various training batches and are then fed into the encoder-decoder model for training the data.

D. Encoder - Decoder Model

After the process of vectorization of words, we further feed these vectors to an Encoder-Decoder Model. Encoder - Decoder Model is an approach which uses sequence 2 sequence mechanism which uses a set of input words sent into the encoder and we obtain an answer or next predicted words through the decoder. To build this encoder-decoder model, we use the concept of LSTM, a special kind of Recurrent Neural Networks (RNN) [16] cells which are capable of learning long-term dependencies. Encoder- Decoder model contains two parts:

1. The encoder which takes the vector representation of the input sequence and maps it to an encoded representation of the input.
2. The final state of the encoder is then fed into the decoder to generate output.

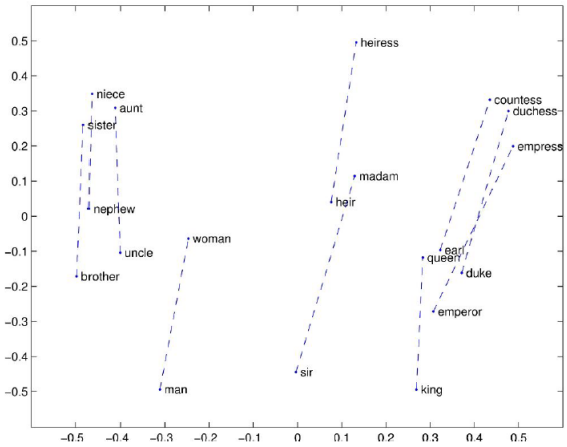


Figure 3: Representation of word embedding

First, let us dwell into the concept of RNN's, it is a machine learning algorithm which is used for analyzing sequential data and is widely used to design many chatbots across the world. Since RNN's have a memory unit they can store important data which helps in remembering important things provided as input to the RNN cell.

As shown in *figure 4*, RNN does not only take inputs sequentially but also work as a loop by taking previous inputs into consideration. RNN's also incorporate the concept of backpropagation which means going back into the neural network which helps in finding the error of partial derivatives with respect to weights and also you can subtract the error value from the actual weights which increase the accuracy of the result.

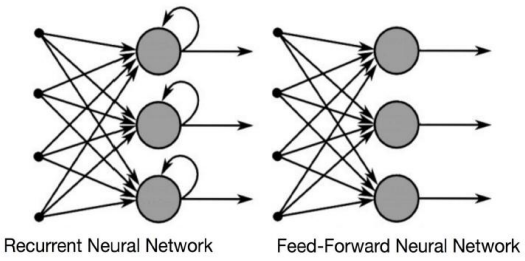


Figure 4: Representation of an RNN

These derivatives can then be used by a Gradient Descent algorithm which is used to minimize a given function. This algorithm keeps on varying the weights either increasing or decreasing in order to minimize the error. Neural networks learn to minimize during the training process.

E. Long Short Term Memory

LSTM networks are an extended version of RNN. LSTM's enable RNN's to store the input information over a long period of time because LSTM can store the information in a memory unit.

This memory unit is referred to as a gated cell where the cell decides to keep or discard the information based on the importance of the information. The importance is determined by the weights which are calculated while training the algorithm. After the training, the network learns which data is important and which can be neglected.

An LSTM cell consists of three gates namely input gate, forget gate and output gate. Input gate determines the amount of new data or input passed into the cell. Forget gate discards all unwanted information. Output gate is used to compute the output provided the values given by the input gate and the forget gate. We can see an illustration of LSTM below in figure 5.

The gates in an LSTM as shown in figure 5 are analog, in the form of sigmoids, they range from 0 to 1. As they are analog, it enables them to do backpropagation with it. LSTM outperforms the other models when we want our model to learn from long term dependencies. LSTM's ability to forget, remember and update the information pushes it one step ahead of RNNs.

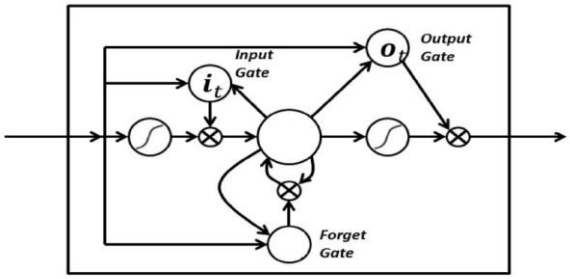


Figure 5: Representation of LSTM

After the vectors from the word embedding are passed to the LSTM network for encoding and decoding, with softmax activation function, rmsprop optimizer, and optimal hyperparameter tuning, we obtain a model which is split into train and test set using the train_test_split function. Further, the model is trained to obtain the required accuracy.

F. Encoder Algorithm

- Each Character from the dataset is encoded in the form of one hot vector which has the length equal to the number of encoder tokens. These characters are provided as input in sequence to the encoder.
- The argument for the return state in the LSTM layer of encoder model is set to true. The hidden state output is also returned by the LSTM layer of encoder model which also contains hidden and state of the cell for all the cells present in that layer. This information is used to define and compute in the decoder model.
- The encoder inputs passed to the encoder to obtain the encoder outputs and the encoder states. 4 The encoder outputs are discarded and only the states are kept

E. Decoder Algorithm

- The response characters which are in one hot vector encoded form are converted into binary vectors which are of length same as the number of decoder tokens. These characters are provided as input to the decoder model.
- The layers of LSTM in the decoder is defined to return the state and the sequences. The hidden and the states of the cells are discarded and only the sequence of output from the hidden states is used for reference.
- We design the decoder to return both internal states and output sequences. In training the model, we
- don't use return states.
- The cell state from the encoder and final hidden units from the encoder are used for initializing the states.
- To predict the characters or produce the characters in the output sequence in a one-shot manner we use a dense output layer. The entire target sequence which is necessary for providing an input to the model is well known during the training of the model.

V. RESULTS

This section presents the results from the perspective of training the model to make it working, graphical user interface for query input and response generation and output interpretation.

Training the Model

We trained the model on a modified gunthercox dataset which includes a set of YML files by default and some of our originally created files as well, with 50 epochs and a batch size of 32, word embedding size was set to 100, we took categorical cross-entropy as our loss function and optimizer used was rmsprop. We got decent results with these parameters.

We trained and tested our model on NVIDIA GTX 1060 Ti. We further plan to improve the accuracy obtained by hyperparameter tuning.

Graphical User Interface

The user can ask a query to the chatbot, which replies to the query. This functionality is provided using a custom GUI developed using Python.

Output

In figure 7 and figure 8 we can see the variation of hyperparameters results in the variation of loss. By visualizing the graphs, we can see that our best results were achieved when our hyperparameters ie. batch size and the number of epochs was set to 16 and 150 respectively.

The generation of a response follows the same procedure that was used to train our model the only exception is that the question asked may or may not be present in the training set, thus if there is a new question response is generated based on

the trained model so at times the response generated could be unsatisfactory.

```
Human:hi
ChatBot:  hello
Human:    How are you
ChatBot:I am doing well
Human:Can I ask you a question
ChatBot:  Sure ask any way
Human:What is your name?
ChatBot:  CAISY
Human:What is your age?
ChatBot:  I am still young
Human:   Tell me a joke
ChatBot:  I will never forget a case, but, in your case I'll make an
exception.
Human:   Who is your boss?
ChatBot:  I like to think myself as self-employed
Human:   What is AI
ChatBot:  Artificial Intelligence is the branch of engineering and science
devoted to construction of artificial ..
Human:   Are you a programmer?
ChatBot:  Of course I am a programmer
Human:   What is chemistry?
ChatBot:  The science of Mixing chemicals
Human:   How far is the moon?
ChatBot:  The moon is 2,50,000 Miles from earth on average.
Human:   Do you like lord of rings?
ChatBot:  My favorite movie is lord of rings
```

Fig 6: Conversation of a normal user with CAISY. The output obtained from CAISY after training the YML format datasets and with hyperparameters set at an optimized value

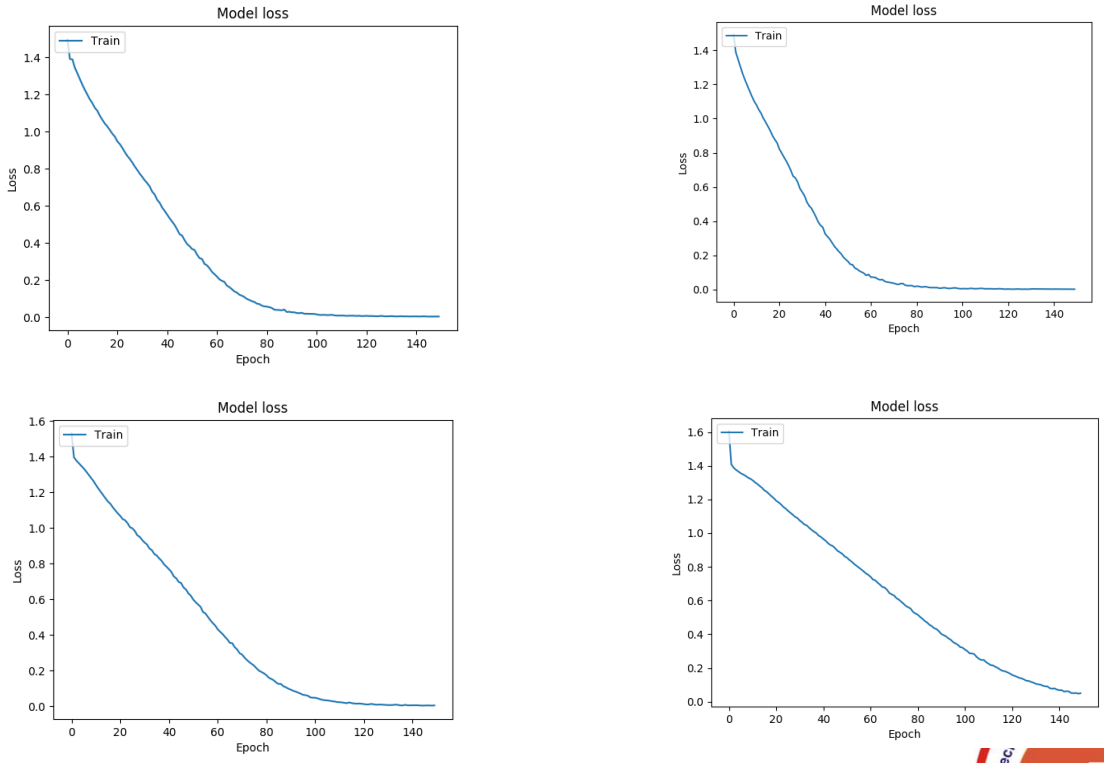
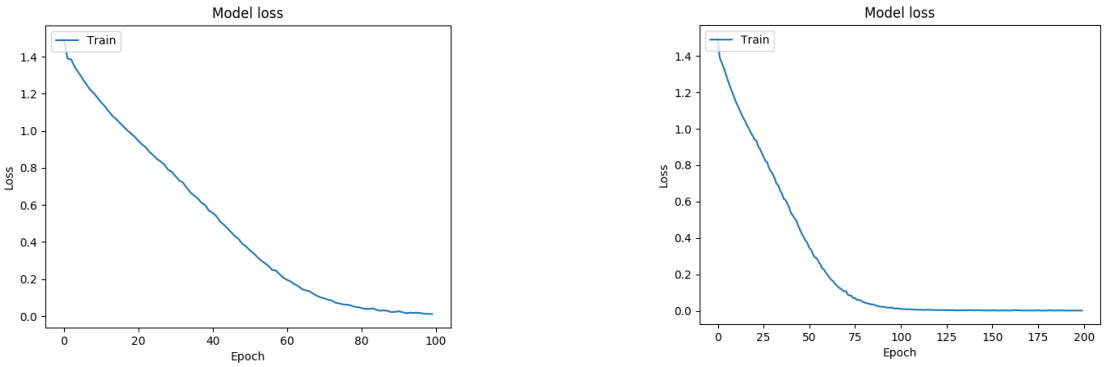


Fig 7: Variation of Batch Size with Epochs as Constant. The Batch Size in this figure are 16,8,32 and 64 respectively



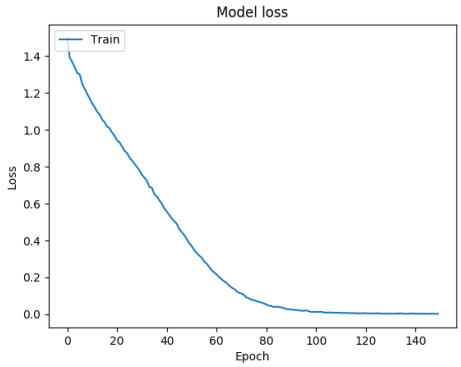


Fig 8: Variation of Epochs with batch size as constant

VI. CONCLUSION

Chatbot can facilitate communication and helps in handling common queries pertaining to specific domain. In this paper, we have demonstrated an open source chatbot framework named CAISY. It can be trained on any dataset of any field/domain to obtain a fully functional chatbot. CAISY uses word embedding, Sequence 2 Sequence model, and Long Short Term Memory neural networks.

CAISY is capable of responding to test queries under a maximum of 5 millisecond delay. We can positively conclude that various domains have a usecase for chatbots, and implementing better, the fully functional chatbot is not difficult anymore with the help of existing technologies. Our continued part in this topic included exploring how attention can be utilized for building an effective chatbot.

REFERENCES

1. Cahn J. CHATBOT: Architecture, design, & development. University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science. 2017 Apr 26.
2. Lei Wang and Ivan Lee, “Artificial intelligence in the 21st century”, March 26, 2018.
3. Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.
4. Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." In Advances in neural information processing systems, pp. 3104-3112. 2014.
5. Perrone, Valerio, Rodolphe Jenatton, Matthias W. Seeger, and Cedric Archambeau. "Scalable hyperparameter transfer learning." In Advances in Neural Information Processing Systems, pp. 6845-6855. 2018.
6. Turing, Alan M. "Computing machinery and intelligence." In Parsing the Turing Test, pp. 23-65. Springer, Dordrecht, 2009.
7. Joseph Weizenbaum, “Computational Linguistics”, Massachusetts Institute of Technology, Cambridge, January 1966.
8. Colby, Kenneth Mark. "Ten criticisms of parry." ACM SIGART Bulletin 48 (1974): 5-9.
9. Deryugina, O. V. "Chatterbots." Scientific and Technical Information Processing 37, no. 2 (2010): 143-147.
10. Perlmutter, D., 2006. Raise a smarter child by kindergarten. New York: Morgan Road..
11. Dasgupta, Dipankar, ed. Artificial immune systems and their applications. Springer Science & Business Media, 2012
12. “Siri”, https://en.wikipedia.org/wiki/Siri
13. Rishab Mehrotra, Ahmed Hassan Awadallah, Ahmed El Kholy, Imed Zitouni “Hey Cortana! Exploring the use cases of aDesktop based Digital Assistant”, Tokyo, Japan, August 2017
14. Hyunji Chung, Michaela Iorga, Jeffrey Voas, “Alexa, Can I Trust You?”.
15. Sepp Hochreiter, Jurgen Schmidhuber “LONG SHORT TERM MEMORY”, Neural Computation 9(8):1735-1780, 1997.
16. Zachary C. Lipton, John Berkowitz, Charles Elkan “A Critical Review of Recurrent Neural Networks for Sequence Learning”

AUTHORS PROFILE

Dr. Manoj Kumar M V, is an Associate Professor**,** in Department of Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru. He has Completed Bachelor of Engineering, and Master of Technology in Computer Science and Engineering from VTU. He holds Ph.D. in Computer Science and Engineering from National Institute of technology Karnataka, Surathkal.He has authored and published and 30+ research articles in reputed international journals and conferences. His research interest includes Machine Learning, Data Mining, Process Mining, and Statistical Data Analysis etc.He is the founder and CEO of Rectopage Software, and Think Learning Foundation, which are functioning successfully from the year 2011.

Retrieval Number: A10210581C19/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Prajwal J M is an undergraduate student currently studying 4th year of Bachelor of Engineering in Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru. His research interest include machine learning and web application development.

Shyamanth Kashyap is an undergraduate student currently studying 4th year of Bachelor of Engineering in Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru.

His research interest includes machine learning, data mining etc.

Rahul G is an undergraduate student currently studying 4th year of Bachelor of Engineering in Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru.

His research interest includes machine learning, data mining, and Mobile Application Development etc.

Pavan Naragund is an undergraduate student currently studying 4th year of Bachelor of Engineering in Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru. His research interest includes machine learning, data mining, and software robot construction.