A Virtual Chatbot for ITSM Application

Neha Atul Godse
Department of Computer Engineering,
Pune Institute of Computer Technology
Pune, India
neha24godse@gmail.com

Shaunak Deodhar
Department of Computer Engineering,
Pune Institute of Computer Technology
Pune, India
shaunakdeodhar996@gmail.com

Shubhangi Raut
Department of Computer Engineering,
Pune Institute of Computer Technology
Pune, India
rshubhangi52@gmail.com

Pranjali Jagdale
Department of Computer Engineering,
Pune Institute of Computer Technology
Pune, India
pranjali.jagdale@gmail.com

***Abstract*—In the current scenario, the end user of Information Technology Service Management application in software companies has to keep on searching the solution for problems he is facing or else finally create a ticket since he cannot collaborate with the system by asking questions and getting relevant answers. As a solution to this, we aim to design a chatbot that will be specifically tailored for software firm employees. The chatbot can process input using Natural Language Processing and can generate a relevant response which will help the end user to solve his query. The chatbot makes decisions itself to answer user's query with the help of the neural network. It will also remember the context of the conversation and perform tasks such as creation of ticket on behalf of the user.**

Keywords— ITSM, chatbot, neural network, NLP

I. Introduction

A chatbot is a software program, which uses Artificial Intelligence to interact with humans. The idea is to simulate human conversations wherein one end is user and another end is a machine. These bots have found applications in various domains such as E-commerce services, medical assistance, recommender systems and educational purposes. They can be integrated into existing application platforms.

ITSM comprises of all the managerial aspects of IT businesses. Nowadays, every software firm has a dedicated team of IT experts which oversee the issues faced by other employees of the company.

In the traditional framework, the end user generates a ticket for any cause, issue or query. This query gets assigned in a queue to some employee at ITSM department. The end user needs to wait until his query gets assigned to ITSM employee. It might take several days because of which he may not be able to proceed with his work. Once the ticket gets assigned, the ITSM employee communicates with the end user employee and solves the issue after which the ticket generated gets dismissed (Refer Figure 1).

In the proposed system, we model a chatbot which can be integrated with any ITSM application. The end user need not generate a ticket every time he faces some issue. Instead, he will be able to have a two-way textual conversation with the chatbot. The input to the chatbot will be in the human spoken language which is processed using Natural Language Processing. NLP is a field of AI that consists of computer understanding and generation of human language. The tokens generated using NLP are then passed to the Neural Network model which is trained using the dataset which consists of conversations. The neural network is a collection of interconnected elements which process information dynamically and generates a response to the external inputs. The output given by the neural network model is passed as the response to the end user. The proposed system is based on the Sequence to Sequence (Seq2Seq) model introduced by Cho Et al [10] which is best suited for conversation modeling and machine translation. Seq2Seq model consists of two Recurrent Neural Networks (RNNs) namely an Encoder and a Decoder. RNNs are the neural networks having loops in them which accept variable length inputs and produce variable length outputs. The information persists in RNN allowing them to use internal memory to process arbitrary sequence of inputs. The encoder converts the input tokens into an intermediate representation consisting of fixed size features known as thought vector or context vector as it denotes the intent of input. The decoder converts the thought vector into words for output response.

The chatbot also remembers the context of the conversation as well as the user and can perform required actions accordingly. The end user can carry forward the chat in this way until his query gets solved. In such a way, the end user can receive a faster response as compared with the traditional framework of ITSM application. If the end user is satisfied with the chatbot, the frequency of tickets generated gets reduced and hence the manpower required at ITSM department will be reduced. Sometimes it may happen that the end user remains unsatisfied with the response given by the chatbot. In such case, the end user can ask the chatbot to generate a ticket on behalf of him.

In this paper, we describe related work including few implemented chatbots in fields other than ITSM. We then describe our proposed system architecture and implementation details.

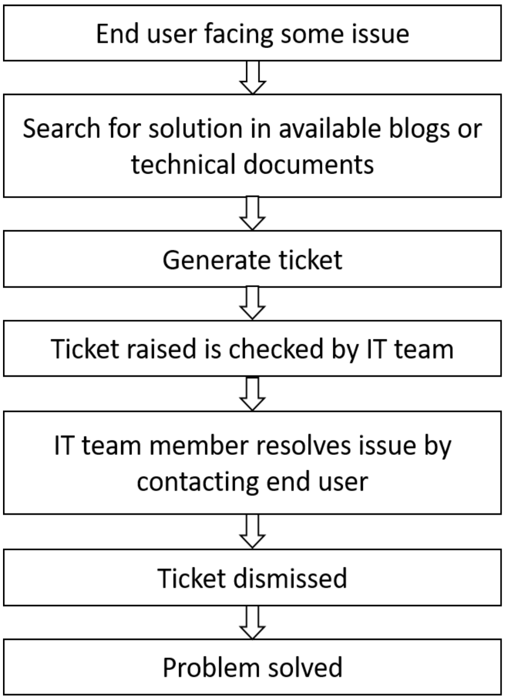

End user facing some issue
Search for solution in available blogs or technical documents
Generate ticket
Ticket raised is checked by IT team
IT team member resolves issue by contacting end user
Ticket dismissed
Problem solved

Fig. 1. Traditional Workflow

II. Related Works

Niranjan Et al [1] have proposed an interactive conversational system for students, using the Naïve Bayesian concept. This model acts as a virtual teacher which can answer the query of the student in an interactive way using the chat agent that is used. Overall system design includes a lexical parser to extract keywords, Bayesian theory for categorization of knowledge, Probability theory for responses.

Thomas N T [2] has proposed a solution for failing cases of Artificial Intelligence Markup Language (AIML) by adding a Latent Semantic analysis (LSA) block in his design. If AIML fails to answer, the question is transferred to LSA block, where it goes through tokenization and stemming. Virtual agents based on AIML have developed domain-specific bots for the user such as Humorist Chatbot System [4], Dorothy Network Management chatbot, Medical assistance virtual agent [5], Tutorbots, Web based Voice bots [6].

A model was presented by Supratip Ghose et al [3] in which they have investigated ALICE chatbot system to serve a domain-specific service, as a UFAQbot. The UFAQbot could act as an advisor to undergraduates who tend to have doubts regarding Universities and academic advice. The main aim of this system is to reduce the efforts of the user to browse through several web pages to retrieve the necessary information. The ALICE system (Artificial Linguistic Internet Chat Entity) uses an XML dialect called AIML [2] [3]. AIML can handle general questions and greetings but fails in the case of unexpected questions.

There has been an evaluation of chatbot system over a period. Old age chatbots had fixed database and could respond only to known queries. In these, PATTERN and TEMPLATE are written such that when a pattern is encountered in user input, one of the responses from the template is given. These chatbots can be called as rule based models or retrieval based models. By using Natural language processing techniques and machine learning algorithms, there has been an attempt to simulate a human-like conversation with chat agents which can learn from existing conversations. This type comes in intelligent models or generative models for a chatbot.

Deep learning has shown promising results in generative models of chatbot than any other handcrafted chatbots. Chatbots that are developed using deep learning, mostly use a certain variant of sequence to sequence (Seq2Seq) model. Seq2Seq model for chatbot approach was proposed by Oriol Et al [7] for conversational modeling. The advantage of this model is that it uses an end-to-end approach that is developing a single model instead of different models for each subtask of chatbot development. The model proposed by Iulian V. Serban Et al [9] extends the hierarchical recurrent encoder-decoder neural network model and improvises it by bootstrapping the learning from word embedding. Further improvements include topic information incorporation into response generation explained by Chen Xing Et al [8].

Some notable developments so far are ELIZA (1966), PARRY (1972) [1], Saya - Japanese chatbot used as a receptionist in a hotel, A.L.I.C.E., Jabberwacky and D.U.D.E (Agence Nationale de la Recherche and CNRS 2006). Not only text-based agents but also voice bots are finding applications in several domains, such as Siri, Google assistant.

III. Proposed System

The workflow of the proposed system is given in Figure 2.

The system receives input from the end user in the form of natural language. This input is bound in JSON object and passed to the NLP module. The JSON object is unbound in the NLP module and further NLP operations are carried out on the input. These operations include tokenization and stemming of the words.

1. Chatbot module: It accepts the input from the end user in the form of natural language. The input is bound in a JSON object and is sent to the NLP and pre-processing module.
2. NLP and pre-processing module: Natural Language Processing module interprets the input to understand its meaning. Also, some preprocessing of input is done in this module. NLP module includes the following operations.
    - Tokenization is the task of slicing text into units known as tokens. These tokens may contain inflected words. Thus, we use stemmer.
    - Stemming is the process of finding root of the derived words which may exist in the tokens. These stemmed words are passed for further data preprocessing.

    Pre-processing of data for training:

    - Padding: As the Seq2Seq model allows to operate on variable length sequence vectors, the preprocessing

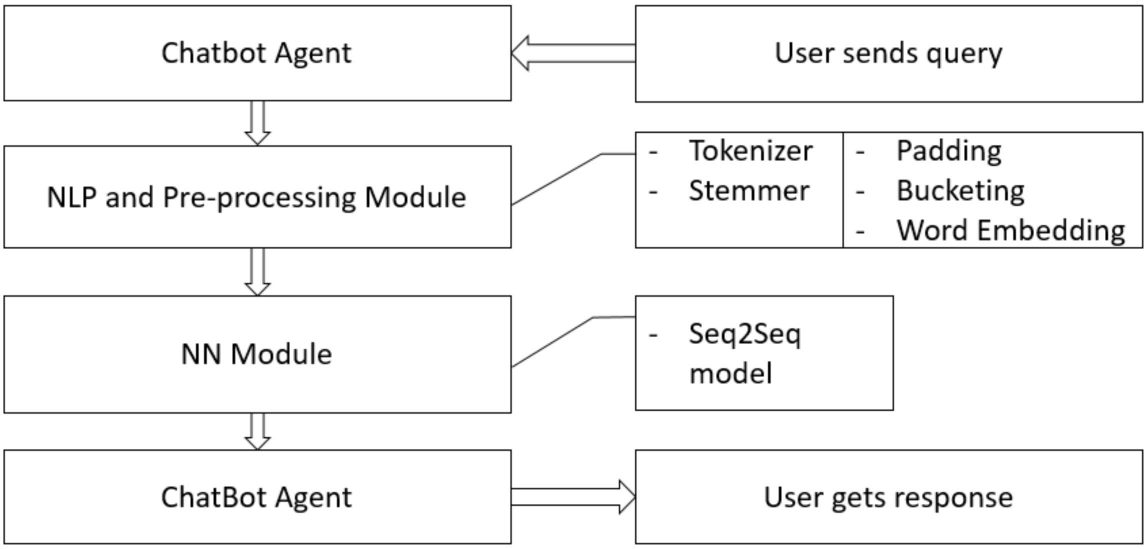


Fig. 2. Proposed System Workflow

needs to be done to convert variable length sequence vectors into fixed length sequence vectors. This is achieved by padding a few special symbols to the sequence vectors.

- Bucketing: In case of too large and too small sentences, padding won't be efficient. Bucketing is used to assign the sentence in one of the buckets according to its size. While training, different model is used for different bucket, keeping the model parameters constant.
- Word embedding: Word2Vec is a model used for word vector initialization of vectors into manageable scale while considering the dimensionality of vectors. This model was proposed by Tomas et al [11] to compute vector representations of words from the dataset.

3. Neural Network Module: This module represents the implementation of Seq2Seq model. The Seq2Seq model is given in Figure 3.

Seq2Seq model consists of two types of RNN namely an Encoder and a Decoder. Given a sequence of inputs X one at a time to encoder, it converts the input to fixed size vector c. Then, the decoder computes the probability of output sequence Y with c as input. Let $X = (x_1, x_2, ..., x_T)$ and $Y = (y_1, y_2, ..., y_{T'})$. Seq2Seq model maximizes the generation probability of Y conditioned on X i.e. $p(y_1, ..., y_{T'} | x_1, ..., x_T)$. Hence the objective function of Seq2Seq can be given as

$$p(y_1, ..., y_{T'} | x_1, ..., x_T) = \prod_{t=1}^{T'} p(y_t | c, y_1, ..., y_{t-1})$$

Steps in LSTM:

1. The information that is unnecessary is discarded in this step. That is decided by the "forget gate layer". This function works on the input value x_t and previous layer's output h_{t-1} to output a number between 0 and 1.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

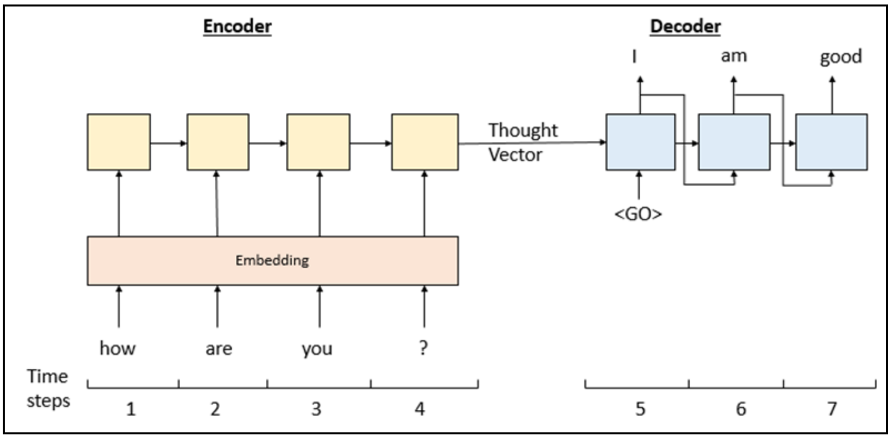


Fig. 3. Seq2Seq Model

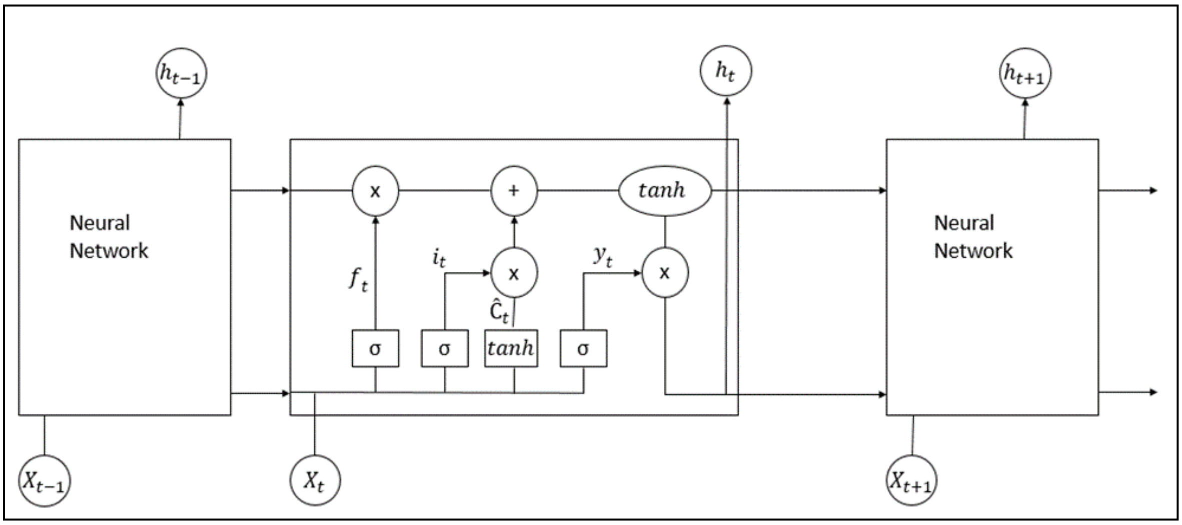


Fig. 4. LSTM Representation

2. This step decides new information that is to be added in the cell state. There are two parts in this:
   a. Values are updated using "input gate layer"

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

   b. *tanh* layer creates a vector of new values to add to the state.
3. Update old cell state C_{t-1} into new C_t. Multiply old state (C_{t-1}) with f_t to discard the things that need to be discarded and add $i_t * \hat{C}$.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$

4. Output is the filtered version of cell state. The cell state output is dependent on the result produced by the sigmoid layer.

$$y_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\hat{C}_t = \tanh(W_i \cdot [h_{t-1}, x_t] + b_C)$$

tanh function is carried out on the cell state and multiplied with the output of sigmoid layer.

$$h_t = y_t * \tanh(C_t)$$

Thus, Seq2Seq model generates the potential response respective to the given input. This response is sent back to the chatbot agent.

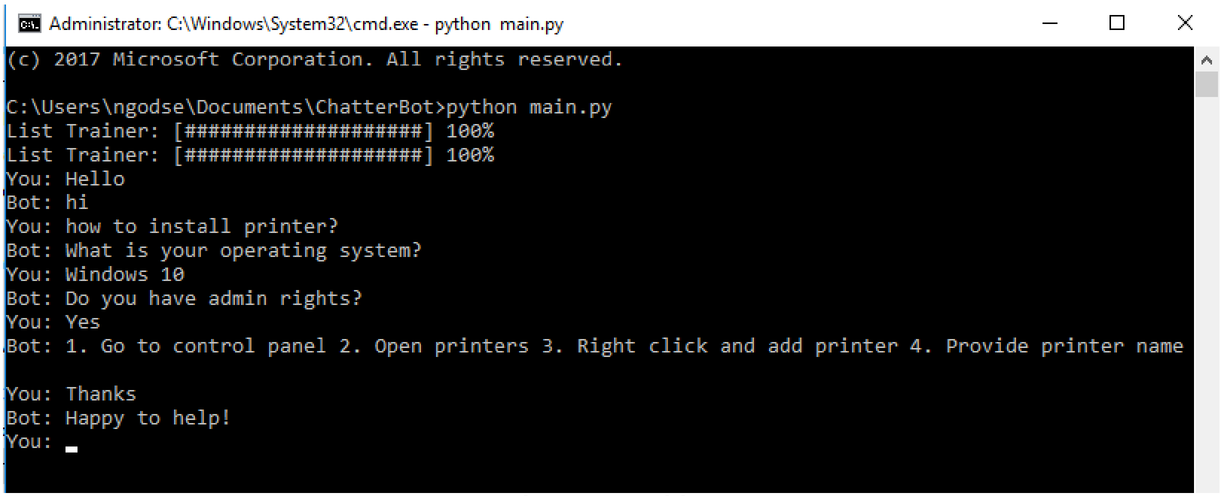
IV. Implementation

A chatbot for ITSM in software companies has been implemented based on the proposed model discussed above. As the chatbot is tailored to be used in software companies, we have considered a virtual office structure, a database of employee details and certain other databases to store the state or context of the system. The designed virtual office demonstrates the location of employee according to designation, the locations of devices that are used nearby such as ACs, coffee machines and printers. This data is further used for querying the system and issue a ticket.

The input is bound in JSON object and through REST API call sent to the chatbot plugin. Chatbot plugin sends the input to the Neural Network Module. After processing the input and generating response, it is captured in chatbot plugin. The response is checked in plugin and modified if necessary. Further, the chat application displays the final response. Refer Figure 4 for a sample implementation module.

The sample use cases implemented are as follows:

1. Installation query(printer/tomcat):
   - Suggest some solutions and generate ticket if query is not solved.
2. Location Change
   - User enters the new location to which he would like to move to.
   - The availability and the post feasibility of the location entered is checked.
   - Finally, the option for auto-allocate or ticket generation is given.
3. Coffee machine not working
   - Coffee machine near the user is identified. If not the same machine, the coffee machine ID is taken from the user.
   - Ticket is generated accordingly.
4. AC: Turn ON/OFF
   - AC near the user is identified.
   - If not the same AC, the other AC ID is taken from the user.
   - AC is turned ON/OFF accordingly.
5. Request Virtual Machine
   - Specifications for the virtual machine are taken from the user and accordingly ticket is generated.
6. Handle irrelevant requests.
   - Ticket is generated if irrelevant request is encountered more than once.



```
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\ngodse\Documents\ChatterBot>python main.py
List Trainer: [####################] 100%
List Trainer: [####################] 100%
You: Hello
Bot: hi
You: how to install printer?
Bot: What is your operating system?
You: Windows 10
Bot: Do you have admin rights?
You: Yes
Bot: 1. Go to control panel 2. Open printers 3. Right click and add printer 4. Provide printer name

You: Thanks
Bot: Happy to help!
You:
```

Fig. 5. Implementation Module

7. VPN not working
   - Suggest some solutions and generate ticket if query is not solved

V. Conclusion

In this model, we proposed a chatbot system using neural networks concept and improve the experience of software firm’s employees. The system would be capable of analyzing natural language input given by the end user and respond back with the solutions generated by neural network model. The chatbot also stores the conversations with each user and learns from them, to provide a better and faster solution if the same problem is faced by a different employee. Further, we could embed the chatbot in ITSM of various software companies.

References

[1] Vincenzo Di Lecce, Marco Calabrese Domenico Soldo, Alessandro Quarto, “Dialogue-Oriented Interface for Linguistic Human-Computer Interaction: a Chat-based Application”, Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2010 IEEE International Conference.

[2] Niranjan.M, Saipreethy.M.S., Gireesh Kumar.T., “An Intelligent Question Answering Conversational Agent using Naïve Bayesian Classifier”, 2012 IEEE International Conference on Technology Enhanced Education (ICTEE).

[3] Supratip Ghose, Jagat Joyti Barua “Toward the implementation of a Topic specific Dialogue based Natural Language Chatbot as an Undergraduate Advisor”, Informatics, Electronics & Vision (ICIEV), 2013 International Conference.

[4] N T Thomas, “An E-business Chatbot using AIML and LSA”, 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

[5] Divya Madhu, Neeraj Jain C. J, Elmy Sebastain, Shinoy Shaji, Anandhu Ajayakumar, “A Novel Approach for Medical Assistance Using Trained Chatbot”, International Conference on Inventive Communication and Computational Technologies (ICICCT 2017).

[6] Bayu Setiaji, Ferry Wahyu Wibowo, “Chatbot Using A Knowledge in Database”, 7th International Conference on Intelligent Systems, Modelling and Simulation, 2016.

[7] Vinyals, O., and Le, Q. V. 2015, “A neural conversational model”, CoRR abs/1506.05869.

[8] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, Wei-Ying Ma, “Topic Aware Neural Response Generation”, CoRR abs/1606.08340.

[9] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, Joelle Pineau, “Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models”, CoRR abs/1507.04808.

[10] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, CoRR abs/1406.1078.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, CoRR abs/1301.3781.

[12] Christopher Olah (2015, August 27). Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding -LSTMs.