An Intelligent Web App Chatbot

SHAZIYA BANU[1]
M.Tech student
School Of Computing and Information Technology,
Reva University,
Bangalore, India.
e-mail: shaziyabanu8888@gmail.com

SHANTALA DEVI PATIL[2]
Professor
School Of Computing and Information Technology,
Reva University,
Bangalore, India.
e-mail: shantaladevipatil@reva.edu.in

Corresponding Author: shaziyabanu8888@gmail.com, mobile no.: +91 – 9742202321.

Abstract— Chatbot is an Artificial Intelligence program that stimulates interactive human conversations with computers. It plays a vital role now-a days due to its 24/7 support for customer queries. Web app chatbot uses a Linguistic machine learning algorithm (NLP) for predicting correct responses for human queries. LUIS (Language Understanding Intelligent Service) is a natural Language processing Artificial Intelligence for predicting human queries. In this paper, chatbot is implemented on LUIS for predicting the user queries. Based on the highest prediction score, Luis detects the intents and entities, to solve user queries. A web app chatbot discussed in this paper is fast, accurate and secure with high performance. Using LUIS API's for automated LUIS training and publishing the endpoint to the chatbot. Enhanced authentication secures the bot from unauthorized persons accessing the chatbot.

Keywords— Chatbot, Natural Language Processing, Language Understanding Intelligent Service, Artificial Intelligence, Enhanced Authentication.

I. INTRODUCTION

Web Application Chatbot is an interactive way to solve user queries. Chatbot is a virtual assistance, conversational interface for humans. It can be developed by NLP, Chatbot based on NLP(natural language processing) are very efficient to use for accurate and high performance. NLP is the brain of the chatbot which thinks and predicts according to human behaviour.

Natural Language Processing (NLP) [1] is a linguistic Artificial Intelligence that manipulates different human languages automatically. It has been defined by modern practitioners to understand the users for better direct the conversation flow. Cognitive service is a set of machine learning algorithms which helps in Artificial Intelligence. Azure Cognitive Service (LUIS) is a service that helps developers build intelligent chatbot.

FIGURE 1: BASIC CHATBOT ARCHITECTURE.

Microsoft Azure cloud service helps the user build a secure web app chatbot. They have been defined by modern practitioners to understand the users for better direct the conversation flow in building a secure bot for enterprise application. Azure bot service helps in naturally interacting with the users by easily integrating cognitive service.

Microsoft Azure provides open SDK tools and build tools to test and publish the bot in different channels for customer interaction. Integrated powerful AI tools for building chatbot like speech, vision, QnA maker, language understanding, search.

Azure Cognitive Service (LUIS) helps in training the chatbot for predicting user intentions while conversation with bot. Integrating powerful AI capabilities with Azure cognitive service. Connecting bot service to utilizing its resources through channels like facebook messenger, slack, skype, microsoft teams, cortona, telegram, website or app.

II. RELATED WORK

Chatbot developed using different languages and based on different technologies such as AIML, ANN, KNN, machine learning algorithms, artificial intelligence, NLP [1]. It can be built on various platforms based on different cognitive services, machine learning algorithms, AIML. AIML is based on XML, hat has to be in a pattern. Cognitive services helps in understanding the human thoughts and predicts the user response through chatbot.

Ramya Ravi [2] methodology on chatbot using AIML (Artificial Intelligent Markup Language) query based chatbot which helps the user to understand the website performance. The AIML is a pattern based where the user needs to type in the specific pattern. The Patterns (questions) are predefined and the user needs to follow the same format. If the pattern matches the template then the response for that query is sent back to the user.

The queries of this chatbot has been categorized based on the type of data such as domain related query,

general queries and none of the above. This makes the user stick to the same pattern. Future enhancement can be done on this by adding the NLP for predicting the user query not by following a specific pattern.

Albert Verasius Dian Sano et al., [3] proposed a chatbot using AGNES Algorithm and Google's DialogFlow as a knowledge base. This chatbot helps tourists to visit in the region. The methodology used involves 3 steps:
Step 1: Data preparation on latitude, longitude and snapshots for clustering.
Step 2: Data mining using AGNES algorithm and parameters used are linkage criteria, measurement type to process the data using AGNES clustering by reading the excel and setting rules.
Step 3: Knowledge base is to store the questions and answers for chatbot to solve the user queries are from DialogFlow framework by Google through intents.

It is used as a tourist chatbot and future work can be implemented by clustering on more regions.

Neelkumar P. Patel et al., [4] proposed UNIBOT (UNIversal chatBOT) using Artificial Intelligence and machine learning algorithms specifically designed for their chatbot . The implementation of the unibot is based on algorithm and designing the database for the chatbot with front-end developed using HTML,CSS, jQuery and Ajax used to call and get responses from PHP files. Database table created for storing questions and answers contains a question and 3 fields for one question with 3 types of answers. Unibot is made for universities segregated according to departmental syllabus, events, admission fee, timetable etc. The algorithm developed for this unibot accept, prose and split the sentence then perform SQL query on the words to pick up the answers for the query. If multiple answers are matched, then options are generated to select one of them. If matching doesn't happen, answers are selected based on keywords. If no result is produced, display's sorry message to the user. Can improve Chatbot's accuracy and query prediction using NLP which helps in predicting the correct response instead of giving multiple options and can save the user's time.

Shafquat Hussain & Prof. Athula Ginige [5] proposed chatbot using AIML and VDMS for Diabetics management system in patience.

In this proposed system when the user registers with the registration form, the data obtained from the user is first normalized. Normalization process then matched with pattern with previous stored responses in database and AIML database, for matched data, response will be displayed to the user else search in the Wikipedia using MediaWiki API plug-in. If the query doesn't match any of the above criteria a default reply is sent to the user.

Amber Nigam et al., [6] proposed chatbot uses Recurrent Neural Network (RNN) model by classifying intents (purpose) and entities (action to be taken for the user query). Proposed model user queries has been categorised to intents and sub-categories using RNNs and entities found at multiple stages of neural network model. The intents and entities are both achieved in parallel for predicting user query. There are 6 steps involved in this model.
Step 1: pre-processing before the first RNN model.
Step 2: predict category of the intent using RNN.
Step 3: Rule base classification for identifying subcategory of intents.
Step 4: Pre-processing before the second RNN model where some entities were replaced with keywords.
Step 5: Using RNN for predicting subcategory of intents.
Step 6: Finding remaining slot-entities using NER.

Weak pattern matching capability can be improvised by using NLP for predicting human behaviour.

Belfin R V et al., [7] proposed a chatbot for cancer patients using NLP, web scraping technologies and database. Database acts as a knowledge base for chatbot.
Step 1: Data Collection: Large amounts of datasets from cancer forums are sorted out and taken in the local database using a beautiful soup library of python and stored in the form of CSV(comma separated values) to the database .
Step 2: Data Pre-processing: Tokenizing the raw data, converting to lowercase, punctuation removal, stop words removal and stemming.
Step 3: In the third step, the processed data is then converted to a graph model using NEO4j in the form of nodes and edges. These graphs help in processing and differentiate and identify the relationship between each data easily.
Step 4: In the last step, based on the symptoms provided to bot by the user is identified and listed the name of cancer based on the classification of the symptoms of cancer and remedies for the cancer and related information is provided along with the treatments to cure cancer.

Without using a Knowledge base, NLP can be used for predicting the human behaviour of asking questions.

Chirstoph Matties et al., [8] proposed a chatbot to record the progress of retrospective action items using Agile Retrospectives chatbot. Slack is used to communicate with the team members as chatbot front-end. This chatbot measures the capability of the employee to work using provided code statements and will inform team members about the status per action item. Based on these results the team members are initiated and act as a basis for data improvements on the sprint board to follow the agile principle. Helpful in following automated agile methodology.

Jitendra Purohit et al., [9] proposed a chatbot for placement drives to interview candidates. Made use of technologies like NLP and decision making. The JARO chatbot is an automated interview process via voice and text. Based on the responses obtained from the candidate using chat conversation, a detailed summary of interview process is generated into a final report and sent to the manager or HR via email. The implementation of chatbot involves JARO (to extract candidate's personal details), JARO chatbot (Based on the details and resume provided to the chatbot. The bot

prepares a set of questions based on the job category selected by the candidate from the database and recorded the response from the candidate), Database (the response is stored in the database along with the questions and scores based on grades) and due to sentimental analysis (the responses are categorized into three levels as low, medium and high) the positive and negative behaviour of the candidate can be predicted.

Advantageous in the education field for candidates interviewed during placement and reduces time.

Sathit Prasomphan [10] proposed chatbot through NLU, ANN model for the process of creating a web application for managing Facebook Page online selling. The proposed model has been segmented into two parts first is System part and second is User part, In System part, products edited through owners page name, information and image. Users can see the entire product list, he can click, upload images and edit information for products and can order products on the list page. Stable libraries can be used by using Reactjs and can improve the performance of chatbot.

Rohit Binu Mathew et al., [11] developed android application chatbot.
Step 1: Chatbot registration.
Step 2: Process messages using NLP for analyzing and predicting user queries and responding to those queries.
Step 3: Trained using KNN for pattern recognition.

The chatbot is made to detect the disease based on the symptoms provided by the patient during conversation, using KNN mapping them to their corresponding diseases. The chatbot answers to the symptoms, future work can be advice for a good consultation.

III. CHALLENGES

A) **Recognizing user intent:** Intent in chatbot represents the objective of the user input. For each user input, an intent must be specified. Recognizing the correct intent for the user query is most important, contrarily the user might end up with getting incorrect response. A challenging task in recognizing intent for user query is designing the structure and classification of intents for the user query. Should have highest prediction score for right intent after training.

B) **Limitations of NLP:** Natural Language processing [1] is not much advanced when it comes with multi linguistic training. NLP has come up with intent entities , sentimental analysis, spell check. But still challenges come with NLP while building a bot in a different language.

C) **User Language:** User language may vary depending on user mood and his requirement. Detecting human emotions in the utterance is a challenge while training the bot.

D) **Understand the user input:** Understanding the user input with different phrases, contexts and meaning varies from person to person. Training the bot with proper intent, entity, utterance is the most important and challenging task , since predicting the user input is difficult because the training score and predicting score of the chatbot should be highly predictive.

E) **User attention for response:** Grabbing the attention of the user is most important by inserting cards and sending correct responses for user query. Hence it's a very challenging task to predict the correct intent and send the correct response to the user.

IV. SYSTEM MODELS

A) **Microsoft Cloud Service:** Microsoft Azure is a cloud computing service [13] provided by Microsoft for building, testing, deploying, and managing enterprise applications. Develop solutions for business. data services for managing data in the cloud, deploying applications. Provides various services like mobile service, computer service, storage service, data management, media services, data monitoring, IOT, machine learning and Artificial Intelligence.

B) **Azure Bot Service:** Microsoft azure bot service [11] offers the developers to add Artificial Intelligence to their bot conversations. They help in building intelligent enterprise bots with control over data and conversations. The platform gives developers an SDK tool to develop an intelligent bot, ability to make smart recommendations, language translations. Azure Bot service provides scalable, integrated connectivity and development service for intelligent bot which can be used to reach out customers through channels.

C) **Azure cognitive Service:** Azure cognitive service [13] brings developers an artificial intelligence for enterprise bot by calling the API to embed the intelligence of speak, hear, understand, see and accelerate decision making for the bot. Deploy cognitive service anywhere from the cloud. Helps in achieving smart decisions faster. extract meaning from unstructured data or text. Integrate speech processing in bot. Analyze contents of data like images, cards, links.

D) **Bot Framework Emulator:** Bot framework emulator [11] is an application which allows developers to test, debug the chatbot build through Bot Framework SDK. chatbot can be tested either locally on machines or remotely through tunnel. The emulator displays the web chat conversations for

testing through localhost and pulls out all the logs in the form of JSON for each request and response of each message.

E) **Enhanced Direct Line Authentication:** Chatbot built with enterprise data needs to be secured only for enterprise users. Since the data involved in enterprise is sensitive and the bot should authenticate each user whenever a user enters the bot for conversation. Direcline is a channel to connect the bot either to the website or app provided by Azure Bot service. Protecting the channel is most important for security considerations. Enhanced DirectLine [14] allows the developer to add trusted origins and secure the bot by passing unique ID and token.

F) **FrontEnd Module:** For web Application developing User interface component for bot users used react component for robustness. To deploy bot into website for enterprise applications, import the directLine from the react library and react webapp component. call the bot using DirectLine.

V. PROPOSED TECHNIQUES

I. Chatbot Architecture and LUIS Training

A) Chatbot

Intelligent chatbot developed through microsoft azure bot service [11].

Step 1: Create a bot using a resource and app service plan. Select a region to launch the web app with resource group, app service plan for messages through app insights.

Step 2: Connect bot to the Artificial Intelligent Cognitive Service Language Understanding (LUIS).

Step 3: Design the classifications of intents, entities and utterances.

Step 4: Train the LUIS, test the trained utterances. see the prediction scores then publish to the endpoint regions.

Step 5: prepare the bot code and handle the intents and entities at backend.

Step 6: Connect the directline channel, enable enhanced authentication, add trusted origins websites for the enterprise application.

Step7: Use Azure bot service endpoints to connect to the bot service from the website, generate a user ID for each conversation and pass unique ID and token of authentication to the direct line to connect to the bot service for conversations.

Chatbot can be connected to various communication channels such as Facebook Messenger, Skype, Cortana, Slack and website (direct line) etc.

B) LUIS

LUIS (Language Understanding Intelligent Service) is a cognitive service for predicting the user query[12]. It's a linguistic machine learning API service for training the bot. Train the chatbot by providing intents, entities and utterance for conversational bot. A conversational bot built with intents, entities and utterance classifying intents, entities and utterance according to the requirement. It gives intelligent algorithms to interpret user needs through understanding using natural methods of communication. A prediction score indicates the degree of confidence LUIS has for prediction results, based on a user utterance. A prediction score is between **zero** (**0**) and **one** (**1**).

Example :

Intent – Greeting
Entity – welcome (classified as simple entity)
Utterance – Hi
Hello
Hey etc
Response – Hello, how may i help you.

II. Automated LUIS Training

LUIS app built with intents, entities and utterances which consists of classified intents, entities and utterances for each intent. Intents define certain actions (predict overall meaning). Entities act as action for the query. Utterance are the user queries which come under each classified intents. The basic intents entities and utterance has been built, trained and published to the endpoints.

Automated LUIS training for inserting the intents, entities and utterance in the LUIS app, training and publishing to the end point region from the admin website. Admin need to add or select the intents in the website form and enter the utterance for that intent, and label the entities using check boxes, then train and publish in just one click through the website. Figure 2 is architecture for automated LUIS training.

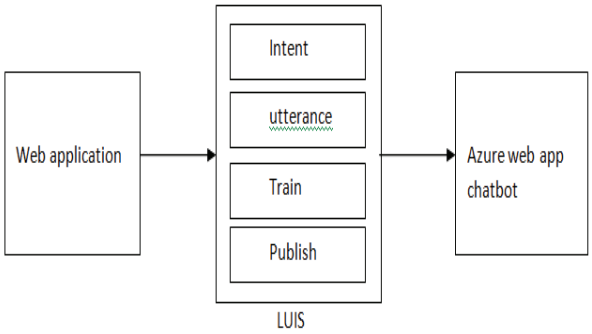

FIGURE 2: AUTOMATION FOR LUIS TRAINING.

III. Chatbot Architecture

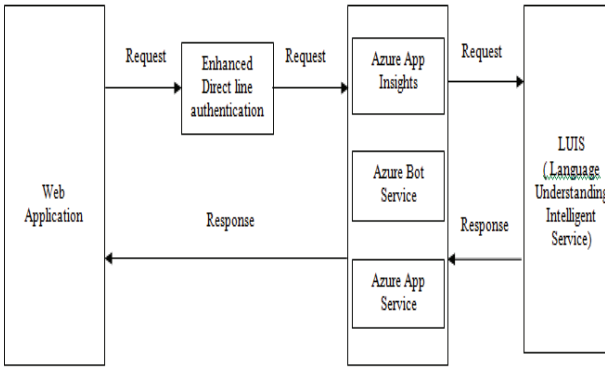

FIGURE 3: AZURE WEBAPP CHATBOT ARCHITECTURE.

Web application emplacing the integrated web app chatbot within the portal. When the user sends a request (user query), the user is authenticated through unique User ID and token for that bot, if token matches the request is sent to the azure bot service, the bot connects to the endpoint of LUIS and predicts the intent, entities based on the highest prediction score of LUIS training for the user entered utterance. The backend code of Azure Bot Service should handle the intents, entities and send the cards as response to the user query either text or link based on the requirement, request and response. Figure 3 shows the azure web app chatbot architecture.

IV. Flow Chart

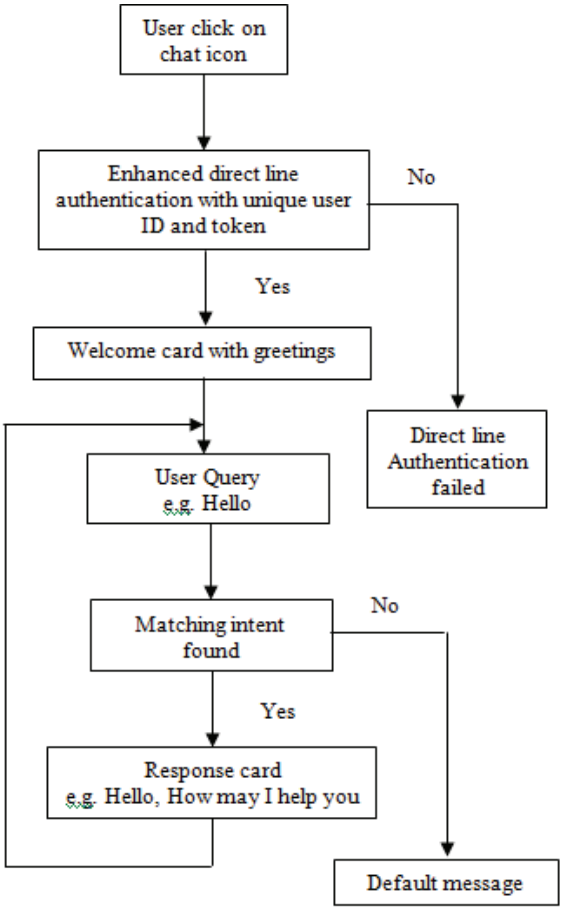

FIGURE 4: FLOW DIAGRAM OF CHATBOT CONVERSATION.

The conversational flow of web app chatbot is shown in figure 4. When the user clicks on a chat icon the bot authenticates the user using the user ID and token then connects to the bot by calling the direct line channel with the token, showing a welcome card with a greeting message and giving on the hint what bot can answer, else it shows a message with direct line connection failed. When the user asks a question to the bot, based on the highest prediction score the intent and entity match, the response is sent to the user with the card of text or link based on the proper answer for the user query. LUIS understands the user query based on the highest prediction score, the response generated for user in the website.

VI. RESULT AND DISCUSSION

The proposed Chatbot is a secured bot for enterprise applications that helps the user solve their doubts related to the portal and gives solutions for their queries. Proposed bot is very accurate and precise and gives the user query response within seconds. Authenticates the

user based on token and unguessable User ID. Authentication and direct line connection all happens within seconds. designed with a welcome card which gives the hint and image of bot to the user. When the user asks the question to the bot, it responds accurately with the correct predicted response since the LUIS classification is done properly with correct prediction and training. LUIS Connection is checked by the chatbot within a fraction of seconds due to the fast connection and response of bot. This is due to the Bot service of Microsoft Azure and due to the LUIS training for predicting the human behaviour during queries generated by the user. Figure 5 shows the screenshot of the response of the chatbot when the user says HI.

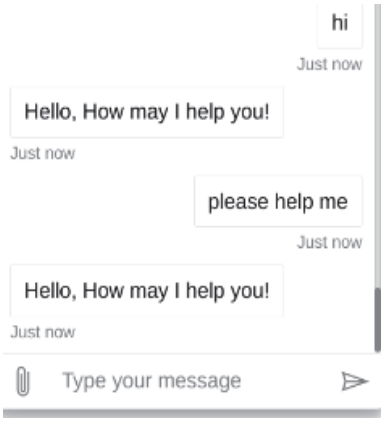

Figure 5: Response for greeting intent.

When the user says bye the bot responds within seconds and greets the user. Figure 6 shows the conversation of the bot and the user at the end of virtual assistance.

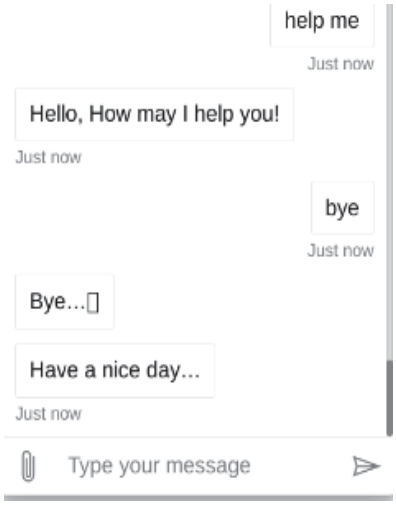

Figure 6: Response at end of conversation.

VII. Comparison Of Existing Techniques

In comparison with existing technology, LUIS gives better results. The results have been compared in Table 1.

Table 1: Comparison of existing chatbot with respect to methodology, intent score, Response and performance.

Sl. No.	Paper Ref No.	Methodology	prediction score	Response of user query	Performance of chatbot
1.	[6]	NLU, Recurrent Neural Network Model¸Bi-LSTM	75.07%	Overall accuracy for 10% bias was 75.07 for intents.	moderate
2.	[2]	AIML	No intent, pattern based matching	Correct response is 65%.	moderate
3.	[3]	AIML	-	Overall response was moderate, response for website is low.	moderate
4.	[4]	AIML	-	Weak pattern matching	moderate
5.	Proposed System	LUIS, Azure bot service	85% to 99% (converted from 0 – 1 scale to percentage)	99% accurate correct response, response generated within seconds	Very Good

VIII. Conclusion and Future Work

In this paper, Microsoft Cloud service platform along with its machine learning AI is used for building the bot. Since it's a bot service within the cloud they provide security options for securing the sensitive data of chatbot from the threat.

Enhanced authentication eliminates the magic code and gives an option to secure the bot by generating a token for the unique user ID using DirectLine channel.

Chatbot developed using Azure bot service gives the quickest response for the user. LUIS predicts the human behaviour during the conversation based on the trained predicted score. Azure provides a cognitive service API for developing the virtual assistance which gives a friendly approach to interact with the user can build with cards in an innovative way.

Automated LUIS training helps in training the bot within fraction of seconds with interacting with the LUIS

app. Through the website in just one click, the admin can insert the intents, utterance, train and publish the bot through the admin login portal.

The proposed system gives the virtual assistance for the user and generates the correct response within seconds. Further, the future work can be implemented by enabling the speech recognition for the bot.

REFERENCES

[1] Vagelis Hristidis, "Chatbot Technologies and Challenges", published in First International Conference on Artificial Intelligence for Industries, pp. **126, 2018.**

[2] Ramya Ravi, "Intelligent Chatbot for Easy Web-Analytics Insights", published in IEEE International Conference, pp.**2194-2195, 2018**.

[3] Albert Verasius Dian Sano, Tanto Daud Imanuel, Mega Intanadias Calista, Hendro Nindito, Andreas Raharto Condrobimo, "The Application of AGNES Algorithm to Optimize Knowledge Base for Tourism Chatbot", published in International Conference on Information Management and Technology (ICIMTech), pp. **65-68, 2018.**

[4] Neelkumar P. Patel, Devangi R. Parikh, Prof. Darshan A. Patel, Prof. Ronak R. "AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT)", published in the Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA], pp**. 148-150, 2019.**

[5] Shafquat Hussain , Prof. Athula Ginige "Extending a conventional chatbot knowledge base to external knowledge source and introducing user based sessions for diabetes education", published in 32nd International Conference on Advanced Information Networking and Applications Workshops, pp. **698-703, 2018.**

[6] Amber Nigam, Prashik Sahare, Kushagra Pandya, "Intent Detection and Slots Prompt in a Closed Domain Chatbot", published in IEEE 13th International Conference on Semantic Computing (ICSC), pp**. 340-343, 2019.**

[7] Belfin R V, Shobana A J, Megha Manilal, Ashly Ann Mathew, Blessy Babu, "A Graph Based Chatbot for Cancer Patients", published in 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. **717-721, 2019.**

[8] Chirstoph Matties, Franziska Dobrigkeit, Guenter Hesse, "An Additional Set of (Automated) Eyes: Chatbots for Agile Retrospectives", published in IEEE/ACM 1st International Workshop on Bots in Software Engineering(BotSE), pp. **34-37, 2019.**

[9] Jitendra Purohit, Aditya Bagwe, Rishabh Mehta, Ojaswini Mangaonkar, Elizabeth George, "Natural Language Processing based Jaro-The Interviewing Chatbot", published in Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC), pp. **134-136, 2019.**

[10] Sathit Prasomphan, "Improvement of Chatbot in Trading System for SMEs by Using Deep Neural Network", published in IEEE 4th International Conference on Cloud Computing and Big Data Analytics, pp. **517-522, 2019.**

[11] Microsoft Azure Documentation, "Azure Bot Service", Azure bot service documentation. [Online] Available: https://docs.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-3.0 . [Accessed**: Jan, 2020].**

[12] Microsoft Azure Documentation, "LUIS", Language Understanding (LUIS) documentation. [Online] Available: https://docs.microsoft.com/en-us/azure/cognitive-services/luis/. [Accessed**: Jan, 2020].**

[13] Microsoft Azure Wiki [Online] Available: https://en.wikipedia.org/wiki/Microsoft_Azure. [Accessed**: Jan, 2020].**

[14] Enhanced Direct Line authentication [Online] Available: https://blog.botframework.com/2018/09/25/enhanced-direct-line-authentication-features/ [Accessed**: Jan, 2020].**