Chatbot Assistant for English as a Second Language Learners

Sanyog Vyawahare
Trainee Software Engineer
Abzooba India Pvt Ltd.
Pune, India
sanvyawahare@gmail.com

Kaustubh Chakradeo
Trainee Software Engineer
SP Innovations and Technologies
Pune, India
chakradeokaustubh@gmail.com

2020 International Conference on Convergence to Digital World - Quo Vadis (ICCDW) | 978-1-7281-4635-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCDW45521.2020.9318672

***Abstract*—This work demonstrates an experimental implementation of a helper bot using IBM Watson. It is primarily aimed at people who know English as a second language. With the help of IBM Watson Assistant tool, the chatbot uses APIs like Google Translate API, Text to Speech API, SimpleWIki and Musixmatch API, to provide features like rich responses, translation to regional languages, text to speech conversion facilities, useful information in simpler English, and displaying music lyrics for music in regional languages. This is particularly helpful for those who are newly learning English and are more comfortable in their regional language.**

Keywords*—*chatbot, NLP, IBM watson assistant, re-gional language.

I. INTRODUCTION

A chatbot is a computer program built in such a way that it tries to mimic human speech. To accurately simulate typed conversation the computer program must understand the user question and must be able to generate the correct response. To accomplish this, Machine Learning plays an important role in building a chatbot using Natural Language Processing. The chatbot can break the user question into intent, that is the purpose of the user and the entity, that is the term or object relevant to the intent. In general, a chatbot is a conversational agent built which interacts with the user on a specific subject. Chatbots can be used in various domains like education, customer service, entertainment etc.

There are various platforms to build a chatbot. Developers must select a platform based on the purpose and the capabilities of their chatbot, like integrating the chatbot agent with NLP for better understanding of the user input. Chatbtots can be divided into 3 categories [1].

1) Non programming chatbots
These platforms are developed for those who have no knowledge or experience of coding. The idea behind these platforms is that one doesn't need to know ML and NLP programming concepts to build their chatbot. Only a simple interactive dialogue flow needs to be built. Some examples of such platforms are Chatfuel, MEOKAY, BotKit, Motion.ai, etc.

2) Conversation Oriented
These platforms are usually built for entertainment purposes, where there is no need to store the context and data of the previous chats. The focus of such a chatbot platform is to create a conversational agent which focuses on user's conversation without getting deep insights and understanding of the user input. The models are generally built using a specific language known as AIML. ALICE and Anna are such examples another example is Mitsuku bot created using AIML technology. Intelligent conversational chatbots built using AIML were studied during the literature review[14][15]. Pandorabot is another platform to build conversational chatbots.

3) *Platform provided by big tech giants.*
Many big tech giants have stepped up into the market with the technologies and ML/NLP integration to make the bot intelligent and to provide users with rich responses. These platforms are easy to use, for creating simple and complex chatbots based on the use case which can be seen in[16]. IBM Watson Assistant, Google's DialogFlow and Facebook's Wit.ai are some of the examples. A chatbot built using this service can be deployed on various communication channels like Messenger, Slack, etc[1].

The section Background Review and Literature Survey is divided into 2 parts, Current chatbot platforms, and Literature survey, which is further divided into Overview of Chatbots and Application Specific Chatbots. This section describes the current frameworks used for building chatbots and studies some papers about chatbots, and particular applications. The next section Implementation describes the overview of the chatbot, including the primary objectives and target audience. It also describes the use of the different APIs in the chatbot and how they are integrated with the Watson Assistant Tool. The next two sections Results and References give the conclusions from building the chatbot and the resources used to create the chatbot.

II. BACKGROUND REVIEW AND LITERATURE SURVEY

A. Current Chatbot Platform

1) DialogFlow
DialogFlow, previously known as Api.ai is a platform

provided by Google for building chatbots. It offers a friendly user interface for anyone to build their chatbot. It can be deployed on various platforms like Google Assistant, Slack, Messenger, Twilio, etc. It supports over 20+ languages. The developer can integrate the chatbot into their application fairly easily by using Kommunicate or they can build their personalized UI. Google also provides a free standard plan for small businesses to build their chatbots.

2) Watson Assistant

IBM Watson Assistant formerly known as Watson Conversation is a Question-Answering(QA) program build by IBM which is capable of understanding Natural Language. It is able to respond to the user without any explicit programming. Using this platform developers can build their chatbots and deploy them across various platforms and social media websites. They can also use Watson API to build and deploy their chatbots on web-based applications. Watson Assistant can also be integrated with external APIs or use IBM's speech to text, text to speech API or Watson discovery etc to improve the understanding of the chatbot.

3) Wit.ai (Now Facebook Owned)

Wit.ai is a natural language interface capable of converting sentences into structured data. It was acquired by Facebook in 2015. Wit.ai can help create bots that interact with humans on messaging platforms. Wit.ai is an open-source platform powering hundreds of apps, wearable devices and healthcare platforms. Earlier, the platform was used only for text-based communication, but since the acquisition by Facebook, it was integrated with Messenger and now includes audio/video communication as well.

4) Microsoft Bot Framework

Microsoft bot framework provides a platform to build and deploy bots on various platforms like Facebook, Skype, etc. This framework can be divided into two parts, either using Azure or Bot Framework as a platform, smartly managed by bot connector and the other in which the bot gains knowledge and deep understanding of users' questions using Microsoft cognitive service called LUIS (Language Understanding Intelligence System). LUIS helps understand the user-input by parsing it into intent and entities. To build a smart chatbot the bot must be able to manage the conversation flow which can be done by properly designing the dialogue flow.

B. Literature Survey

Overview of Chatbots*:* AM Rahman et al in their paper[1] studied the working and implementation of chatbots in different domains along with their architecture. They also studied the challenges chatbots currently face and its future prospective. The two important challenges mentioned in the paper are in NLP and ML design and development. Chatbots are mainly built using conversational dialogue engine written in python and the paper shows the working of Chatterbot (python module for chatbot). They further classified the chatbot platform in three different categories, Non Programming chatbots where there is a minimal requirement of programming skills, Conversational-Oriented chatbots where the communication is more proactive, where AIML language is used. Finally, they studied various platforms by big tech giants like Google's Api.ai and Facebook's Wit.ai where the former uses intent concept and later uses the entity-based concept.

Sarthak V. Doshi et al in their paper [2] built an android application chatbot using the open-source program-o AIML interpreter. Using program-o interpreter they built two types of chatbot one text-based and other voice-based and also demonstrated their approach by developing an android application. In their chatbot, if the user input is voice it is first converted into text using speech recognizer API and then passed on to the Program-o API for querying and generating a reply. This response generation phase consists of two phases, preparation of pattern matching and pattern matching behaviour where they tried to find the largest matching pattern for the input. The reply is parsed in JSON format and send to the user. If the response needed conversion into voice they used a simple text-to-speech API for the same. With the use of external API for sport, weather and Wikipedia the chatbot can be made more intelligent.

In the paper [3], Marilyn Choque-Díaz, Jimmy Armas-Aguirre built a cognitive technology model to enhance academic support services. Their aim was to add customer experience patterns to improve interaction with the bot. The bot worked in 5 stages. The first one was Capture- It took input from the user. The next stage was Understanding Natural Language in which the text was converted into structured data. The third stage consisted of Dialogue Management, in which with the help of IBM Watson workspace, the dialogue was converted to business dialogue and CX patterns through a rule engine. The next stage was Generating Responses which was a functional block that sent messages back to the user. The last stage was Consumption, present throughout the flow of the model, to help with the integration with the cloud database. The chatbot was built using IBM Cloud using IBM Watson implemented via Conversion API. It achieved a reduction by 99% in the average response time per query. However, it only worked for text-based chatbot.

Application specific chatbots*:* Nudtaporn Rosruen and Taweesak Samanchuen, in their paper[4], built a medical consultant system service by using chatbot technology. The medical consultant system, called MedBot was developed using Dialogflow by Google's Machine Learning. It covered knowledge about 16 medical symptoms like fever, cough, vomit, rash, stomachache, dengue, etc. These symptoms were used to train the bot which resulted in 34 intents, 1 for each symptom, 5 sub-intents for types of stomachache, 1 for greeting the user, 1 for finding the nearest hospital and 1 for no illness. The symptoms were studied with the help of DoctorME application. The chatbot gave information about

medicines for the symptoms as a part of the output. Api.ai by Google was used as the engine for the Chatbot.

In the paper[5], Albert Verasius Dian Sano, Tanto Daud Imanuel, et al build a chatbot which would respond to tourists' questions about the tourist places in and around a particular city, by applying hierarchical clustering analysis on a set of tourist sites based on Agglomerative Nesting (AGNES) algorithm. The authors listed 39 popular sites and their geolocation calculated by finding out the latitude and longitude. The AGNES algorithm worked by clustering similar sites by using the geolocation as a measure of clustering. Thus clusters of tourist sites were formed based on the relative distances to each other. The results were obtained in the form of a dendogram, which is a tree-like structure. The chatbot could answer questions like, "Which tourist spots could I cover in a one-day trip?" Thus, the authors created a chatbot which could provide answers to tourists which could help them plan their trips.

In their paper[6], authors Tussanai Parthornratt, Pasd Putthapipat, et al demonstrate an experimental implementation of a Smart Home Automation System using Raspberry Pi, Facebook Chatbot and Google Maps API. It provides additional features like Estimated Time of Arrival of the owner, interactive chat and secure communication. They used the readymade Facebook chatbot API platform instead of creating one from scratch. They also included security services like SSL, penetration testing and WebSockets. The Raspberry Pi was the key component used as the gateway operation to the user's home. The user could tell the bot to switch ON and OFF the various systems in the house. However, the main disadvantage of this system was limited use. Also, it required constant Internet connectivity for communication between the server and Home automation. One major disadvantage was the necessity of a Facebook account to use the chatbot. User's not having a social media account on Facebook could not make use of the chatbot services.

In their paper[7], Seshadri Padmanabha Venkatagiri, Aditya K Sinha, et al demonstrate Puppeteer, a decentralized autonomous multi behavioural platform which provides seamless coordination between toys, and a declarative mechanism to author and adapt learning scenarios for enactment. Puppeteer supports a declarative method for specifying social learning scenarios by incorporating temporal logic operators within a probabilistic logic programming framework to represent causality and interdependence amongst the actors' behaviours. Their Architecture is divided into UI Agent and Computer Agent. UI Agent facilitates assigning scenarios, personae, play content and receive input. Controller agent pushes a publisher-subscriber model and a rules engine which choreographs scenario enactment. They also demonstrate the performance of the bot through various simulations to show persona assignments and system response time.

III. SYSTEM ARCHITECTURE

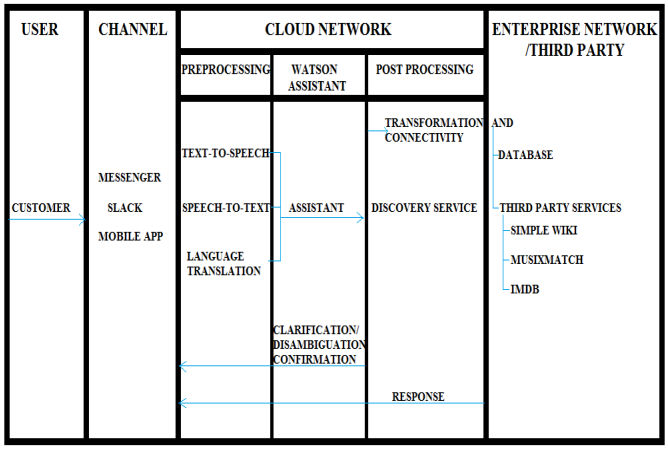

Fig. 1 Chatbot Architecture

Fig 1 describes the architecture of the chatbot[8].

The system is built of the three sections as seen in Fig. 1, the user and channel sections consist the frontend of the chatbot which can be a web application or it could also be deployed on Messenger or Slack channel.

The middle section is the most important one. It is called as the application layer. The user's input is processed in this section. The Watson assistant takes the input and process it to understand the intent and entity to provide the user with the appropriate response. The Watson assistant can also be integrated with some additional Watson cloud APIs such as text-to-speech or language translator. This section can be scaled to use some external services as per the needs or requirements.

The final section is the external APIs or database which can be integrated with the chatbot. This layer is called the post-processing layer. User data can be stored in the database for future use. The bot can be integrated with other external APIs. In this paper, the authors have integrated Simple Wikipedia, Musicmatch and IMDB to enhance chatbot response with the use of these services.

A. Chatbot Design

The chatbot is implemented using IBM Watson Assistant Tool. Once the user starts conversing with the bot, the message is transferred between services in JSON format, to understand the input from the user and produce appropriate responses. The message is initially processed by the Watson Assistant. The bot is also integrated with text-to-speech service to transform audio into text and translation services to convert the bot response into user understandable language. Any conversation may give one of three outputs:

- Successful: A conversation will be considered successful if the assistant can answer with accuracy, giving the user the information he was looking for.
- Unsuccessful: If the assistant can respond to the user demands, but the information given is either irrelevant to the matter or wrong. A conversation will also be considered unsuccessful if the assistant is unable to provide an answer to a question or gets stuck and is unable to carry on with the conversation.
- Undefined: If the conversation ends but it is unclear whether or not the conversation was successful.

IV. SYSTEM IMPLEMENTATION

A. Chatbot Development

To build a chatbot using IBM Watson Assistant, The first step is to create an IBM Bluemix account, and then select the Watson assistant service. This will bring up the dashboard for the Waston Assistant. On this dashboard, there are three fields-intents, entities and dialog. Based on the purpose of the chatbot the intents and entities must be specified accordingly. The bot will understand user input based on intent and entities. Next step is to create a dialog flow, the dialog matches the intent to the corresponding response. Dialog flow is like a tree-build of various dialog nodes each having a set of conditions. Based on the context and intent of the input; it provides the user with a response. Once the chatbot is ready with the flow it can be integrated with other API's to enhance his knowledge base. The chatbot then can be deployed with its very own UI or on platforms like Slack, Facebook Messenger, etc. This can be done by building a client application[9].

In this paper,we created a chatbot for people having English as their second language. The goal of this chatbot is to provide the user with an application through which it can interact and find out the answers to his questions. The answers provided will be either fetched from simple Wikipedia: a website for children and adult learning English. Other API's can also be used to fetch responses. The answer will have simple English words and grammar. If the user wants the answer to be converted into his native language this can also be done by integrating the chatbot with language translator API. Watson provides an API for this but according to the study done for this research, Google Translation is way more advanced and accurate than Watson Language Translator.

B. External Integration

The authors have added a layer of interesting API's on top of their chatbot. The most important API used is Google's Cloud Translation API[11]. This will translate any text that the user wishes, into a regional language, or English from a regional language. Since the chatbot is primarily aimed at people learning the language, it is a very important feature. One of the other APIs introduced in the chatbot was the Musixmatch API[12], which contains information and lyrics about English and regional songs. This is an important feature, since new learners are more accustomed to the songs from their regional regions, and might become interested in English songs as well while learning the language.

Similar to the Musixmatch API, one more important API included in the chatbot is the IMDb API[13]. It is an unofficial Internet Movie Database API, which contains information about all movies in multiple languages. This is another useful feature for English as a Second Language (ESL) learners. Other APIs include text to speech conversion, which combines with the Google cloud translation API. This means that the user can avail the services of translation and audio output at the same time. Overall, the chatbot aims to be a friendly helper to ESL users, with entertainment and learning combined.

Keeping in mind the learners who are new to the English language, the chatbot makes use of SimpleWiki API. The Simplewiki is a version of Wikipedia which is helpful for people who are learning English. The articles use a simpler language and grammar with short sentences. By removing the overly complicated words, the SimpleWiki becomes a useful tool for teaching English language and structure. This API gives any information that the user wishes to peruse, in simple English sentences, thus making his task easier.[10]

V. CONCLUSIONS

In this work, the development of a chatbot helping users learning English as a Second Language was done. The chatbot can translate text into regional languages, give audio output, and look up information for regional movies and music. The chatbot uses simple English vocabulary by making use of Simplewiki, thus making the user's tasks easier, and helping them learn English in the process. The chatbot can respond to user queries with proper answers and also finds additional information requested by the user. This system helps users with ESL by providing them with educational and entertainment content.

REFERENCES

[1] A. M. Rahman, A. A. Mamun and A. Islam, "Programming challenges of chatbot: Current and future prospective," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp. 75-78.

[2] Doshi, Sarthak & B. Pawar, Suprabha & G. Shelar, Akshay & S. Kulkarni, Shraddha. (2017). Artificial Intelligence Chatbot in Android System using Open Source Program-O. IJARCCE. 6. 816-821. 10.17148/IJARCCE.2017.64151.

[3] M. Choque-Díaz, J. Armas-Aguirre and P. Shiguihara-Juárez, "Cognitive technology model to enhanced academic support services with chatbots," 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, 2018, pp. 1-4.

[4] Rosruen, N., & Samanchuen, T. (2018). Chatbot Utilization for Medical Consultant System. 2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), 1-5.

[5] A. V. Dian Sano, T. Daud Imanuel, M. Intanadias Calista, H. Nindito and A. Raharto Condrobimo, "The Application of AGNES Algorithm to Optimize Knowledge Base for Tourism Chatbot," 2018 International Conference on Information Management and Technology (ICIMTech), Jakarta, 2018, pp. 65-68

[6] T. Parthornratt, D. Kitsawat, P. Putthapipat and P. Koronjaruwat, "A Smart Home Automation Via Facebook Chatbot and Raspberry Pi," 2018 2nd International Conference on Engineering Innovation (ICEI), Bangkok, 2018, pp. 52-56.

[7] S. P. Venkatagiri, A. K. Sinha, R. K. Yandrapally, P. Dey and B. Sengupta, "Puppeteer: De-centralized platform for connected-yet-autonomous educational toys," 2018 10th International Conference on Communication Systems and Networks (COMSNETS), Bengaluru, 2018, pp. 49-56.

[8] Medium. Integrate Watson Assistant With Just About Anything (2018). https://medium.com/ibm-watson/integrate-watson-assistant-with-just-about-anything-695bc1d29875

[9] IBM Cloud. Getting started with Watson Assistant (2019). https://cloud.ibm.com/docs/services/assistant?topic=assistant-getting-started

[10] MediaWiki Action API. Overview of MediaWiki APIs(2019). https://www.mediawiki.org/wiki/API:Main_page
[11] Google Cloud APIs. APIs and References — Cloud Translations. (2018) https://cloud.google.com/translate/docs/apis
[12] Rapid API. Musixmatch API Directory (2019). https://rapidapi.com/musixmatch.com/api/musixmatch
[13] Rapid API. IMDb API Documentation(2019). https://rapidapi.com/apidojo/api/imdb8
[14] Ly Pichponreay, Jin-Hyuk Kim, Chi-Hwan Choi, Kyung-Hee Lee and Wan-Sup Cho, "Smart answering Chatbot based on OCR and Overgenerating Transformations and Ranking," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, 2016, pp. 1002-1005.
[15] Ravi, Ramya. (2018). Intelligent Chatbot for Easy Web-Analytics Insights. 2193-2+195. 10.1109/ICACCI.2018.8554577.
[16] P. R. Telang, A. K. Kalia, M. Vukovic, R. Pandita and M. P. Singh, "A Conceptual Framework for Engineering Chatbots," in IEEE Internet Computing, vol. 22, no. 6, pp. 54-59, 1 Nov.-Dec. 2018.