

Artificial Intelligence based Travel Bot

Prof. Firoz Khan, Shrigowri. V. B , Swetha K M , Syeda Ayesha, Vinutha K

Department of Information Science and Engineering,
GM Institute of technology,
Davangere

Abstract:- This paper shows the implementation of an artificial intelligent voice bot with whom human can interact by speaking to it and receive a response by voice bot using its speech synthesizer. Objective of this paper is to show application of voice bot that can be used to book railway ticket using voice commands. It is statistical model and voice bot is based on AIML (Artificial Intelligent Markup Language) structure for training the model and uses Google voice synthesizer for providing speech recognition system and natural language processing. And voice bot is a conversational agent where a computer program is designed to simulate an intelligent conversation. It can take user input in many formats like text, voice, etc. For this purpose, many open source platforms are available. Artificial Intelligence Markup Language (AIML) is derived from Extensible Markup Language (XML) which is used to build up a conversational agent (voice bot) artificially. In this paper, we use 'program o' which is an AIML interpreter for the generation of the responses of users input. We have used this method for developing an android application voice bot which will interact with user using voice responses.

Keywords: *voice bot, speech recognition, AIML, intelligent system, natural language processing.*

INTRODUCTION

What's a TRAVEL BOT??

TRAVEL + CHATBOT = TRAVEL BOT

A Travel Bot is basically a computer program intended to simulate an intelligent conversation with one or more human users by means of textual or aural approaches merely for the purpose of Travel. Most Travel bots today exploit instant messaging as the application interface and mimic human chat exchanges. The key technique of operation of these Travel bots encompass the recognition of cue words or phrases that the user types in and responds with pre-programmed replies that can move the chat forward in a deceptively eloquent manner. To be precise, in reality the present-day Travel bots aren't Artificially Intelligent instead they are just a different user interface built as a messaging thread. Travel Bots can be categorized into the following

CUSTOMER CARE TRAVELBOT: A very common type of travel bot is the customer care bot that is embedded into a travel booking website. These bots are rather obsolete. They are a long way from keen and may summon a similar sort of fierceness that you get when you are on the telephone with a robotized client benefit framework for 60 minutes and you just need to converse with a person. All these travel bots work in analogous ways. They are for the most part restricted in their capacity to help a client book

travel. However they can help clients explore an organization's website productively and also answer some basic queries.

FACE & SLACK TRAVELBOT: These travel bots are by and large for travel search organizations, for example, Sky scanner and Expedia, and these chat bots are somewhat more quick witted, as in they really help you book travel through chat conversation as opposed to simply demonstrating to you where to go on a site to book.

VIRTUAL TRAVEL ASSISTANTS: Though these Travel Bots adopt similar instant messaging interface, they have more features integrated making them the smartest class of Travel bots

Over the last few years, voice bots have played a prominent role as human-computer interfaces. Voice bots are generally composed of three modules: the user interface, an interpreter, and a knowledge base. Laven [6] defines voice bots as a program that attempts to simulate typed conversation, with the aim of at least temporarily fooling the human into thinking they were talking to another person. Basically, voice bots is a conversational agent that can interact with user in a given subject using the natural language. Many voice bots have been deployed on the internet for the purpose of education, customer service site, guidance, entertainment. Existing famous voice bots are ALICE [2], SimSimi and Cleverbot. Artificial Intelligence Markup Language (AIML) is derived from extensible Markup language (XML) which is used to build up a conversational agent artificially. The AIML based voice bots are famous because they are light weighted, easy to configure as well as at minimum cost. AIML has class of data objects called AIML object which describes behavior of computer programs.

In our paper, we have used program-o [1] which is an open source AIML engine written in PHP. It is an interpreter for the AIML scripts of the voice bots. It uses MySQL database to store the chat bot details. Also, we are storing all the AIML scripts to the database. When user sends message to the chat bot program, then according to matched reply from the AIML, the answer is formulated and send back to the user. It can be directly installed on a local server under the GNU General Public License. The voice bots that have been deployed on the internet uses text, voice as well as sentiments as the input. In this paper, we have used the text and voice as user input. Text I/O is relatively effective as user can review for the input so that it can be rechecked if there are any mistakes. However,

giving text input consumes time. So, the solution is to introducing voice interface with the speech recognition technology. By these methods, this voice bots application is very able to make conversation with the user

The below fig shows the base interface of our voice bot:

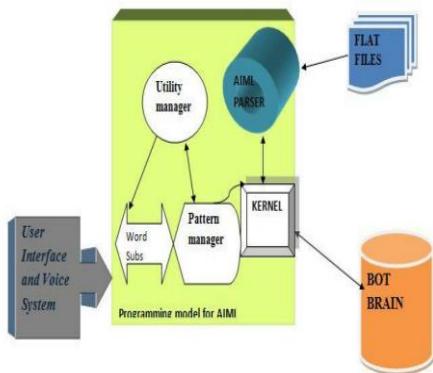


Figure 1: Voice bot interface

Bot Engine

The bot engine is the main architecture which consists of following components like

- Kernel
- AIML Parser
- Utility Manager
- Word Substitution
- Pattern Manager

Kernel

Kernel class has most of the implementation and is responsible for all processing. It requires all other classes for the processing to happen, it require AIML parser, word substitution, utility manager and pattern manager for matching. It include following modules like sessions, bot predicates, word substitute, element processor. It also include brain file , if brain file is provided kernel try to load the brain and if brain file is not their then it attempts to load all AIML files.

AIML Parser

AIML parser is just like xml parser or say more particular it is sax parser. First need to get name of current unknown element. And set skip current category to true and also count the number of errors occurred in the AIML document to test the authenticity of document, if the number of errors occurred are greater than number of threshold value skipped that AIML file.

Utility Manager

It includes the assorted utility function which are used in the AIML package for performing various functions. Its task is to split a given string into list of sentences. Therefore it helps in splitting string and performing all the

required processing by AIML parser or that is sax parser to do work.

Word Substitution

It implements word substitution class adopted from python cookbook module for simplicity and accuracy. This class is used like a dictionary to add pairs of word anywhere in dictionary before or after. It also contain another method called as substitution method for substituting words in the string. All matching done for replacing is done case sensitive and intelligent.

Pattern Manager

This class implements pattern matching algorithm of AIML which is makes pattern matching more intelligent and accurate. The algorithm follows like this:

- Calculate no. of templates currently stored.
- Set the name of chatter bot for its uniqueness and name must consist of single word.
- If any pattern found, dump it for debugging purpose.
- If any further pattern is found store it in a file specified like a pickle file.
- After all patterns are stored, restored all patterns for making a brain map
- After creation of brain , it can used to interact with user.

OBJECTIVES

The main objective of our paper is to built a robust scalable voice bot application which helps the users to book railway ticket online with voice commands.

SYSTEM ARCHITECTURE

The following figure gives the information of our system architecture.

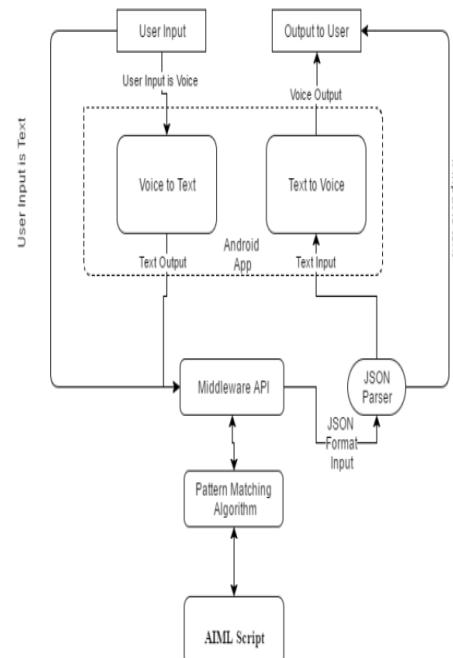


Figure 2: System Architecture

The system works in two modes, text and voice.

When user gives the input in text format the first mode is activated. The user input is passed to the middleware API for the response. On other hand when user gives the voice input then second mode is activated, In this voice mode we first convert the voice into text before sending it to middleware API.

Middleware is the model which connects the AIML scripts with our android app. When user input is received at the middleware, it is passed to the pattern matching algorithm which runs over the AIML scripts. In this process, firstly the pattern matching algorithm is executed for matching of the valid response from the available AIML scripts. When pattern is matched, the corresponding template is returned to the middleware. Then Middleware encodes the template into the JSON format and sends the reply to the android app. After receiving the response app decode the JSON and gives the response to the user.

The response generation process is carried out with two phases [2].

A. Preparation of Pattern Matching

Each input to the AIML interpreter is passed through two main phases.

- Normalization Process for input.
- Producing input path for each sentence.

B. Pattern Matching Behavior

Here we try to find the largest matching pattern and best one by word by word matching of the input. This behavior can be described with Graph master set of files and directories containing a set of nodes which is called node master and branches represents first words of all patterns.

LITERATURE REVIEWS

The bot [1, 2] system began in early nineteenth century where first chess playing machine was built such that a system act as a player and other player an human can play against it, such that system giving a feeling of another person existence. Here is different chatter bot build for various purposes.

ALICE [7-9] was implemented with pattern matching algorithm which was as simple as string matching technique. ALICE takes the text as input and produces output as text which was like question and answer based system. Whereas another chatter bot[5][6] which was build earlier known as Elizabeth requires set of input rules, keyword patterns for identifying input and output rules to produce required response. The recursion [3, 4] based approach was implemented in ALICE chatter bot and it is a vital feature for the model to work properly. Since it was recursion based so calling function by itself made it easy to respond appropriately but it overflow the stack. However rules in other bot may lead to language specific chatter bot such that it sticks to particular language therefore to make

it language independent one has revised all the rules to work it properly and produce output.

ALICE can combine two answers in the case of splitting during the normalization process, or by recursive process. In Elizabeth [5-6] we cannot partition the sentence in two parts and then combine the result. Pattern-managing matching process is the vital feature of ALICE chatter bot, but it is simple and used algorithm is depth-first search which result in producing no output also. DFS tries to find algorithm finds the longest common subsequence pattern matching the required text in the database where other one Elizabeth produce output according to the first keyword matched. Both systems are memory based. Elizabeth's dynamic process, it can perform other actions while other conversation is in process, is considered a excellent feature which is not in ALICE. If the inputs are repeated during the conversation, Elizabeth gives different answers by use of different random selection responses from the list. ALICE use random selection which can lead to duplicate responses.

Since Elizabeth is based on specific language , therefore rules written for it also language based, therefore it is difficult to other language user to interact. However language independent was ALICE but it was string matching based which was not fruitful. Elizabeth requires large amount of analysis tables for the steps to perform in matching a text or string, therefore for the developer how answer question is carried out. From the above it can be understood that none of the previous bot was perfect such that it raises a need of new chatter bot which is language independent and it does not entitle to input a text by writing it has to produce output by just voice enabled and user doesn't has to read also it should also be voice response such that it becomes user friendly for user and it ease of use should it uniqueness.

METHODOLOGY

This section is divided into two parts, first detailed implementation of text based chat bot, second detailed of the implementation of voice based chat bot.

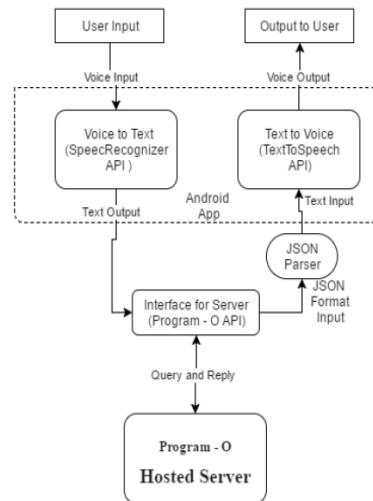


Figure 3: Voice Based Chat bot

A. Implementation of Voice based Chat bot

This approach starts with voice input for the user. Here on start app will ask user for the voice input for user. Then we need to convert the voice value to the text because we have the server with our program – o which understands only here we are giving the voice output. Bot's response is converted to voice and played through the phones speakers. Repeat the steps for continuous chatting with chat bot.

The following steps explain the implementation details to accomplish the above-mentioned tasks:

- Promote user for the voice input on onCreate method. Record the user voice till the speech end event.
- Now convert the voice into text using the Google's Speech Recognizer API of the android [10]. After getting result of the conversion send the data to the server.
- Use the HttpURLConnection in the Async Task for requesting the reply from the server.
- We will get the reply in the JSON Format Parse it to get the reply of the bot.
- Here we are implementing the voice based chat bot so reply from the bot must be voice. Convert the bot's reply to the voice using Text to Speech API for the android .
- After complete play of voice, again ask user for new voice input and repeat the steps.

RESULTS AND DISCUSSION

To demonstrate our approach, we have developed an android application. This application can be used to chat with the bot through voice.



Figure 4: Home Screen

text values. After the conversion of the voice to text the value is sent to the server for the reply. As server replies in JSON format, so we need to parse it. Get the response of the bot after the parsing. We have taken the voice input so,

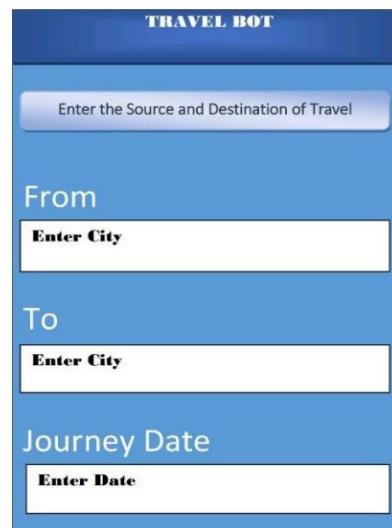


Figure 5: chat using voice module



Figure 6: Travel bot response

CONCLUSION

In this paper, we have introduced a chat bot application in android which is able to interact with users. This chat bot can answer for queries in the textual as well as in voice form of user input. For this purpose, AIML with program-o has been used. The chat bot can answer only those questions which he has the answer in its dataset. So, to increase the knowledge of the chat bot, we can add the APIs of Wikipedia, Weather Forecasting Department, Sports, News, Government Services and a lot more. In such cases, the user will be able to talk and interact with the chat bot in any domain. Using the APIs like Weather, Sports, News and Government Services, the chat bot will be able to answer the questions outside of its dataset and which are currently happening in the real world.

REFERENCES

- [1] Weizenbaum, J. ELIZA – A computer program for the study of natural language communication between 10(8):36–45 (1966).
- [2] Shawar BA, Atwell E, “A comparison between Alice and Elizabeth chatbot system,”University of Leeds, School of Computing research report 2002.19
- [3] Bayan AbuShawar, Eric Atwell, “ALICE Chatbot: Trials and Outputs,”Computación y Sistemas, Vol. 19, No. 4, 2015, pp. 625–632
- [4] Thomas N. T., Amrita Vishwa, “An E-business Chatbot using AIML and LSA,”2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 21-24, 2016, Jaipur, India
- [5] Shawar BA, Atwell E, “A comparison between Alice and Elizabeth chatbot system,”University of Leeds, School of Computing research report 2002.19
- [6] S. Laven (2013), “The Simon Lave Page”, Available at: <http://www.simonlaven.com>
- [7] Md. Shahriare Satu, Md. Hasnat Parvez, “Review of integrated applications with AIML based chatbot,”1 st International Conference on Computer & Information Engineering, Nov 2015,Dept. of CSE, Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
- [8] Rushabh Jain, Burhanuddin Lokhandwala, “Android based Chat-Bot,”International Journal of Computer Applications (0975 – 8887) Volume