International Conference on Advancements in Computing Technologies - ICACT 2018 ISSN: 2454-4248
Volume: 4 Issue: 2 31 – 34

An Overview of the Architecture for Developing an Extensible Personal Assistant Chatbot

S. Geethanjali[1], Birunda Antoinette Mary J.[2]

P.G. Scholar, Department of Computer Science,Stella Maris College, Chennai, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Science,Stella Maris College, Chennai, Tamil Nadu, India[2]

Abstract:In the promising area of Artificial Intelligence, Chatbots are finding more pivotal roles to serve mankind. A Chatbot is a computer program written to conduct conversations with human users using natural language. These Chatbots must be designed in such a way that the users must feel, they are conversing with another fellow human and not a machine. The most anticipative application of Chatbots is in the field of Personal Assistance. This paper proposes a generic architecture for the creation of an intelligent Chatbot, that serves as a Personal Assistant to the user and whose capabilities can be extended. The bot is capable of providing many services, out of which the key role discussed in this paper is, to set-up meetings via e-mail communication.

Keywords:*Chatbot; Artificial Intelligence; AI; Personal Assistant; AIML; RiveScript; Python; API; XML.*

*****

I. INTRODUCTION

Artificial Intelligence (AI) gives machines the power to behave like humans. AI enables the machines to make decisions based on gathered data. An intelligent agent must be able to reproduce the ***cognitive functions*** accomplished by humans, which includes: learning, reasoning, recognizing context, perception, linguistic intelligence and problem solving. A **Chatbot or Chatterbot or Talkbot**is a conversational program that engages with humans using natural language. Communication can either be textual or auditory.

Chatbots were initially developed using pattern-matching techniques, which is still prevalent. Developing Chatbots is possible using scripting languages, mark-up languages, machine learning etc. Each method is subject to certain advantages and disadvantages. Additionally, Chatbots are given features to make them more user-friendly and attractive. This includes adding personality, memory and other such attributes. These features make Chatbots an ideal communication tool that can be deployed across various fields, but not limited to Social media, Healthcare, Education, Customer service, Environmental science etc.

With the advent of technology, more tasks are being automated. One among them is personal assistance. Virtualizing personal assistance is the process of automating the tasks performed by a typical personal assistant. This includes dealing with client correspondence, organizing meetings, managing the user's diary etc., to name a few. A Chatbot would be an ideal virtual personal assistant, as it takes care of the basic functionality required of an assistant – communication. In this paper, the architecture for developing one such personal assistant Chatbot is proposed. In the current prototype, the bot will be responsible for the most basic task of setting up meetings for the user. The proposed architecture is extensible, so that additional tasks can be integrated with ease.

II. LITERATURE SURVEY

The history of Chatbots began with **ELIZA** [1], a pattern matching bot, developed by Weizenbaum in 1964, which takes after a psychotherapist. The big revolution came about with **A.L.I.C.E. (Artificial Linguistic Internet Computer Entity)** [2], a Loebner Prize winner, developed by Wallace in 1995. Wallace introduced the concept of **AIML (Artificial Intelligence Mark-up Language)**. **Jabberwacky** [3], developed by Carpenter took Chatbots one level up by integrating deep learning and fuzzy logic.

A number of real-time Chatbots have since been developed, each serving a different purpose such as virtual dietician, conversation companion, healthcare assistant, personal assistant, career counsellor, shopping guide etc. The following section is aimed to divulge more knowledge about Chatbots through a survey of few Chatbots.

Haller and Rebedea [4] designed a **Chatbot to simulate historical figures**, including their personality, using DBpedia and ChatScript. DBpedia is used to extract data from the web pertaining to the character that the user selects. ChatScript, a scripting language is used to design the conversational agent. The agent's personality reflects the historic character selected by the user. This method provides accurate details and can be used in schools for Computer Supported Collaborative Learning (CSCL). This work can be extended to other important characters in the future.

A pediatric generic medicine consultant Chatbot called as **Pharmabot** was developed by **Comendador et al.** [5] using C# and MS Access. This bot prescribes, suggests and gives information on generic medicines for children, based on its knowledge-base which is populated with details given by pediatricians and pharmacy students. The bot got a good reception and can be extended to include more data.

In their paper, **Gupta et al.** [6] developed a **Chatbot that acts as an online automated assistant**. The bot can easily interact with e-commerce websites to provide product suggestions to users. HTML/CSS, PHP and MySQL are used to develop the website, whereas, the Chatbot was developed using RiveScript. The Chatbot takes in user requirements and provides suggestions.

Chappie, a service provider Chatbot designed by **Behera** [7] is claimed to be a semi-automatic intelligent agent. Chappie works using intent extraction; response generation and conversation routing to a human agent. Naïve Bayes classifier and AIML are used for these purposes respectively. Chappie had an affirmative response.

Mhatre et al. [8] created a Chatbot called **Donna, a personal assistant**. Donna is tasked to set-up meetings for users. Donna's effectiveness comes from the fact that it uses Natural Language Processing to obtain information from e-mails and fix appointments accordingly. Donna also uses Gmail API and Calendar API. QA pattern matching is used to teach Donna to reply to particular questions. The QA patterns are stored in a database. The major disadvantage of this method is that, the QA patterns for all possible ways of conveying the same meaning, has to be input to train Donna. This incurs performance overhead.

III. PROPOSED SYSTEM

The goal of the proposed system is to develop an efficient personal assistant Chatbot, which can communicate with human users as well as other computer programs to accomplish its task. A good personal assistant must be able to perform multiple chores to satisfy the user's needs. It goes without saying that many different APIs will have to be integrated, to accomplish various tasks. An intelligent system must be able to recognize the commands users send to it, though it may be in different forms, and provide an adequate reply. Besides, the system must know its limitations – what it can and cannot do. Furthermore, training the bot's reply must be made more generic and limited to key words, rather than sentences. In order to impart this quality into a bot, this paper proposes the use of an intermediate **translator** module as shown below in figure 3.1.

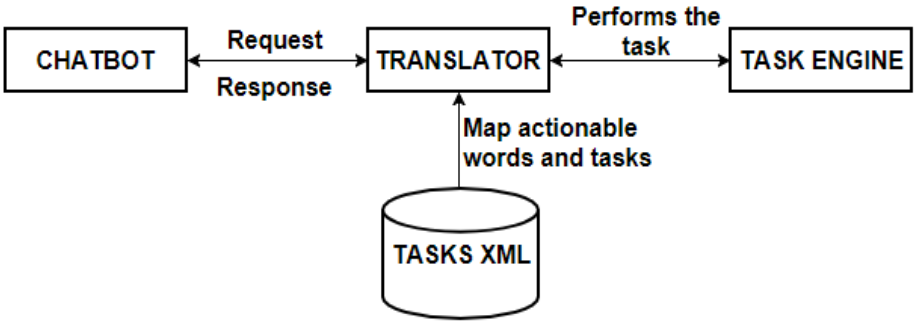

Figure 3.1 Architecture of proposed system

The ***Chatbot*** is the communication agent which acts as the interface between user and the programs. It has its own knowledge-base for generating responses.

The ***Translator*** is an intermediate program that acts as an interface between the bot and the tasks it can carry out. It can check a data store for possible instructions to be followed for any action requested by the user.

The ***Tasks XML*** is a data storage module that keeps track of what tasks have been configured for the bot to work with. It sends an acknowledgement to the translator, which then passes the information to the Chatbot. If the requested action can be performed, required parameters are collected from the user and are passed as a call to the corresponding task engine. In cases where the requested action has not been programmed, the bot conveys to the user that the requested action cannot be fulfilled. The XML file will look like the sample shown below.

```
<tasks>
<task filename="chkMail.py">
<action> read mail </action>
<action> check mail </action>
<action> show mails </action>
..
..
</task>
..
..
</tasks>
```

The ***Task Engine*** module comprises of various programs that offer a range of functions to be executed as the user requests it. The result is returned to the user via the Chatbot interface.

Using the translator gives us an edge over traditional QA pattern matching algorithms because we only need to specify the key words used to represent an action. These words are easily mapped to the tasks. Training the bot would involve mapping of any number of keywords with the actions. This makes this architecture extensible. Likewise searching the XML data store and finding the task does not incur much cost as it would in matching entire QA patterns. Also, the translator will only be accessed when the task requested by the user is not fulfilled by the bot's knowledge-base. Hence the slight overhead that results from checking the XML data store is minimal

IV. METHODOLOGY

For the purpose of developing a prototype, the Chatbot will be tasked with setting-up meetings with clients using e-mail. This works using the following modules:

User Interface:

UI is used to communicate with the Chatbot. The UI can be text-based (GUI or command line) or voice-based. The main function of the UI module is to take in messages from the user and produce appropriate responses. The key challenge in this module is the conversion of auditory inputs into textual inputs with maximum accuracy. The UI must be appealing and

 ISSN: 2454-4248
31 – 34

user-friendly. UI can be developed as a stand-alone or a web application.

Chatbot:

The Chatbot is the most crucial part of the system. It acts as an interface between the user and the APIs. The Chatbot takes the role of a personal assistant and therefore has the following functionality: checking availability of requested function; providing replies; sending appropriate acknowledgment to the respective parties. The knowledgebase of the Chatbot must be designed to facilitate the aforementioned activities. Apart from these activities, the Chatbot must extract and classify the intent of the message received. The user can interact with the Chatbot using natural language.

Translator:

This module acts as the backbone of the entire system. It integrates all the other modules and provides communication channels between them. This performs calls to different APIs as well as to the data store.

Gmail API:

The Gmail API is a RESTful API used to access the user's mail [9]. The API provides access to the mail server, which enables retrieving and composing mails. Authentication processes are taken care by the API. The main form of communication with the client is through e-mail. Hence this module is used to facilitate the purpose of reading, composing and sending mail to clients.

Calendar API:

The Calendar API is used to create, view and modify different events on the user's calendar [10]. This API also uses RESTful calls. The calendar is capable of sharing events, sending reminders and notifications to users about upcoming events marked in the calendar. In this system, the user's schedule is available on the calendar. The translator will check to see if the user is available for a particular time slot and fix up a meeting accordingly. The calendar is updated to reflect the newly added event and to provide reminders and notifications.

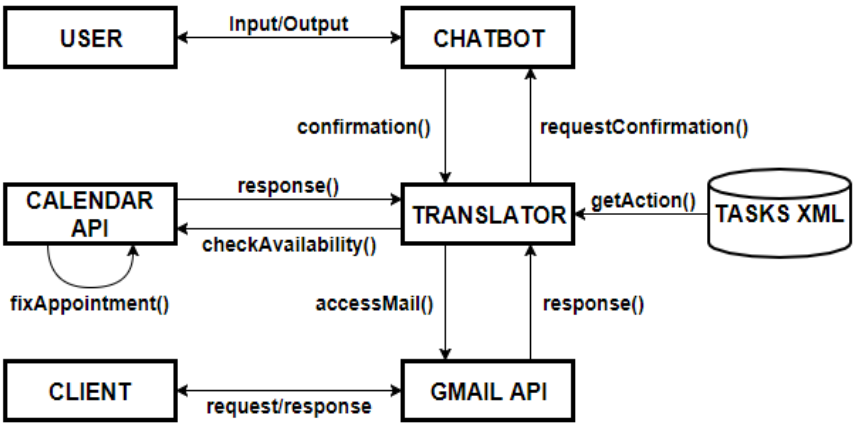

Figure 4.1 Working of the proposed Chatbot prototype

The working of the proposed Chatbot is depicted in figure 4.1. When a client wants to fix an appointment with the user, they send a mail to the user requesting an appointment. The bot is used to monitor the user's mail by looking for unread mails with specific headers. The NLP functions of the Chatbot are used to read the mail and extract required information such as date and time of appointment, place of meeting, client details and the reason for meeting. The translator then checks the user's calendar for the specified time slot. If the slot is available, it sends a confirmation request to the user. Upon the user confirmation, the appointment is fixed and an event is created in the calendar. Following this, an acknowledgement mail is sent to the client. The user can also initiate meetings by asking the bot to send a mail to the respective client. Response from the client is checked for replies and the appointment is fixed.

V. PRELIMINARY SUPPOSITIONS AND IMPLICATIONS

The proposed Chatbot prototype is intended to be implemented using the following tools: RiveScript, Python, Gmail API and Calendar API.

Amongst several factors to be considered in determining the quality of a Chatbot, this paper takes into account, the quality attributes articulated for Chatbots and conversational agents by Radziwill& Benton – which uses the ISO 9241 concept of usability: "The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments." [12].

The accuracy with which the Chatbot provides replies can be calculated based on the number of correct matches performed. The accuracy will be high owing to the features of RiveScript [11], which provides powerful and flexible pattern matching, thus making the bot effective.

Consistency of the bot is with regards to its actions with external APIs. Calls to these APIs must be designed to be robust and the corresponding action must be atomic in nature. This is achieved by writing macros in RiveScript to call python functions to perform calls to APIs and writing code to handle any exceptions that might arise. Use of built-in macros makes the system efficient.

Speed of response also plays a major role in determining the efficiency of the Chatbot. In this case, speed greatly

ISSN: 2454-4248
31 – 34

depends on the internet speed, the processing speed of python files and the size of mails received.

Hence it can be implied that the proposed bot will function as an efficient and effective personal assistant. But the ultimate appraisal of the Chatbot is subject to the user's satisfaction – based on the behaviour, accessibility and personality of the Chatbot.

VI. CONCLUSION

This paper proposes an architecture for the development of an extensible Chatbot that would act as a virtual personal assistant to the user. The concept of using an intermediate translator and a XML data store module will equip intelligence to the bot, since the bot is able to distinguish the tasks correctly and know whether it can provide that service or not. For the purpose of demonstration, the prototype will be given the functionality of setting up meetings. The meeting can either be initiated by the client through a mail to the user or by the user. The Chatbot will access external APIs to accomplish its task. This bot is anticipated to be more efficient, effective and satisfactory in terms of accuracy, consistency and reliability.

REFERENCES

[1] Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), (pp. 36-45).

[2] Wallace, R. S. (2007). Chapter 00. The anatomy of ALICE. Alice AI Foundation.

[3] About the Jabberwacky AI. [Online]. Retrieved from http://www.jabberwacky.com/j2about (Date last accessed 02-Sep-2017).

[4] Haller, E., &Rebedea, T. (2013, May). Designing a chat-bot that simulates an historical figure. In Control Systems and Computer Science (CSCS), 2013 19th International Conference on (pp. 582-589). IEEE.

[5] Comendador, B. E. V., Francisco, B. M. B., Medenilla, J. S., & Mae, S. (2015). Pharmabot: a pediatric generic medicine consultant chatbot. Journal of Automation and Control Engineering Vol, 3(2).

[6] Gupta, S., Borkar, D., De Mello, C., &Patil, S. (2015). An E-Commerce Website based Chatbot. International Journal of Computer Science and Information Technologies, 6(2), 1483-1485.

[7] Behera, B. (2016). Chappie-A semi-automatic intelligent Chatbot. Retrieved from https://www.cse.iitb.ac.in

[8] Mhatre, N., Motani, K., Shah, M., & Mali, S. (2016). Donna interactive Chat-bot acting as a personal assistant. International Journal of Computer Applications, 140(10).

[9] Gmail API. [Online]. Retrieved from https://developers.google.com/gmail/api/ (date last accessed 20-dec-2017).

[10] Google Calendar API. [Online]. Retrieved from https://developers.google.com/google-apps/calendar (date last accessed 20-dec-2017).

[11] RiveScript [Online]. Retrieved from https://www.rivescript.com/ (date last accessed 12-jan-2018).

[12] Radziwill, N. M., & Benton, M. C. (2017). Evaluating Quality of Chatbots and Intelligent Conversational Agents. arXiv preprint arXiv:1704.04579