Donna Interactive Chat-bot acting as a Personal Assistant

Namita Mhatre
Student, K.J. Somaiya College of Engineering. Mumbai- 77

Karan Motani
Student, K.J. Somaiya College of Engineering. Mumbai- 77

Maitri Shah
Student, K.J. Somaiya College of Engineering. Mumbai- 77

Swati Mali
Mentor, K.J. Somaiya College of Engineering. Mumbai- 77

ABSTRACT
Chat-bots are computer programs coded to have a textual or verbal conversation which is logical or intelligent. Chat-bots are designed to make humans believe that they are talking to a human; but instead they are in fact talking to a machine. Taking advantage of this transparency property of chat-bot, an artificial character and personality can be given to a chat-bot which acts like a person of a specific profession. This paper describes an approach to the idea of implementing web-based artificially intelligent chat-bot as a personal assistant of the user, which stimulates setting and initiating meetings of user with his clients. The exchange of information happens through email conversations whereas its evaluation happens through natural language procession and natural language generation and AIML files.

General Terms
AIML, Artificial Intelligence, Pattern Matching.

Keywords
Chat-bot, AIML, Artificial Intelligence, Machine Learning, Google API, AI agent for automated meeting planning, Scheduler, Personal Assistant.

1. INTRODUCTION
Every human on the earth is bestowed with equal amount of time i.e. 24 hours per day. Time is one of the only resource which is fairly distributed among all humans irrespective of their gender, geographical location, caste or religion, educational qualification etc. However, some people reach pinnacles of success whereas others often remain too engrossed in their mundane activities with no time left for something extra-ordinary. One of the most important quality that distinguishes former from the latter is their ability to manage time.

We live in a world of scalability and collaboration. Our work takes us to different places and interact with different people. With every professional having a busy schedule, it becomes really complex to set up an appointment with another individual. If not worked upon the schedule efficiently, it can lead to delay in the meeting which may affect your business proceedings. Failing to plan is planning to fail. But, since we are already short of time, we definitely don't want to spend our time planning if a machine can do that for us!

Here is where DONNA, a web-based personal assistant chat-bot comes into picture. A chat-bot is a program that has the ability to simulate a mundane conversation with a human either via textual or auditory methods [1]. Most of the chat-bots are designed for engaging in small talk and their personalities are created by the programmer. Designing chat-bots using the current state of the art, mainly uses rules written in AIML (Artificial Intelligence Markup Language) or ChatScript. Generally they are implemented to cover a wide range of issues and topics, but also leaving aside many more opportunity areas.

This paper focuses on creating an AI implemented chat-bot which acts like a personal assistant to set a user's meeting with his colleagues or friends through email conversations. It reads the appointment request email from the client with the help of its natural language processing algorithms like pattern matching. From the info obtained, it checks user's availability at the given date & time from user's google calendar. Accordingly, it generates a reply by natural language generation to send it to user. Through several such interactions and correspondence emails it finally fixes a meeting and makes its entry in user's google calendar. It can also initiate a meeting request and follow the same procedure later.

Section 2 of this paper gives an essence of the past work in this domain. The proposed system is thoroughly explained in section 3. The architecture diagram and the implementation of the system are described in section by 4 and 5 respectively; followed by the experiments conducted and their results in section 6 and 7.

Since, this process is automated, it not only saves user's money for hiring a personal assistant but also saves a lot of time and turns out to be very performance efficient and productive.

2. RELATED WORK
The article published by Alan Turing in 1950 proposed Turing test which acts as a criterion for intelligence. The criterion depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge, sufficiently well that the judge is unable to distinguish reliably (based on the conversational content alone) between the program and a real human [2].

People have started implementing different applications of chatbot since long back. One can find numerous real-time chatbots with different personality attributes and different purposes. There exist applications like: a chatbot acting as an undergraduate advisor [3], a historical character [4], a customer service and support assistant [5], an individual on social network [6] etc. A detailed study was conducted on the chat-bot application of a historical character which is described as follows.

Literature survey of Chat-bot as Historic Figure:

Many interactive chat-bots have been developed to stimulate human-computer intelligent interactions. Most of them rely on the database for generation of responses. However, this paper proposes an idea of giving the personality of a real historic figure through the knowledge of the human figure available.

This knowledge is extracted from Wikipedia which stores information in the form of chapter-wise plain texts and DBpedia. The most important life events and relations of the historic figures are taken into consideration for building the desired personality. The two important steps for its execution are:-

I. Extraction of text.

For extracting facts from text, the Stanford CoreNLP libraries are used, which provide a set of natural language analysis tools which can take raw English language text input and perform lemmatization, POS tagging, markup the structure of sentences in terms of phrases and word dependencies, and many other facilities.

A. Using information from DBPedia:

DBpedia is a community effort to extract structured data from Wikipedia and to make this information available on the Web.

B. Extracting Information from Wikipedia:

For this paper, the example of Adolf Hitler is considered. Few important topics from Wikipedia page are taken such as About World War I : Join, Responsibilities, Role, Decorations, Battles, Ideology, and even about Family, Health, Death.

C. Transforming verbs within associated texts from 3rd person to 1st person singular:

i. Replacing all references to character's name with surname

ii. Replacing relevant co-referential pronouns with surname

iii. Replacing possessive pronouns related to the character

iv. Modifying verb form

v. Replacing surname with pronouns

vi. Replacing possessive pronouns

II. Designing the conversation

Designing the conversational agent is not built from scratch. An open-source software called ChatScript is used. ChatScript is a scripting language designed to accept user text input and generate a text response. The program inputs one or more sentences from the user and outputs one or more sentences back.

Additionally, there are some existing systems from which the idea of creating an application of chatbot (virtual personal assistant) was inspired.

1. A.L.I.C.E.

One of the most famous chatbot which works on Pattern Matching Strategy is the Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) [7]. The AIML files for A.L.I.C.E. are available online which contain categories like music, art, philosophy, etc. So for the basic working of Donna, these AIML files are being used. Also, another original AIML file for the category "Meetings" has been generated, which answers specific meeting related questions. Thus, as the project concentrates more towards the scheduling module, AIML files are being used for the pattern matching framework of Donna.

2. x.ai (AMY)

AMY is a personal assistant chat-bot designed by a company called x.ai. It is still in its Beta version and only provides service to some users as of now. A user has to CC (send a carbon copy) of the email to AMY (amy@x.ai). Then, Amy talks to the other party, sets up a preferable meeting timing, and then sends an acknowledgement emails to both the parties [8]. The idea to create Donna was inspired by Amy. Donna is better than Amy in a way that emails have to be forwarded as a carbon copy to Amy; whereas Donna intelligently extracts emails which are related to meetings from the user's account. As Amy is still in its Beta version, it is not possible to compare its working and its features with Donna.

3. PROPOSED SYSTEM

The system extracts the appointment related email from the user account automatically by matching some keywords from the email body and subject. It then reads the email with the help of algorithms like pattern matching. From the information obtained from user's calendar, it checks user's availability at the given date & time from user's google calendar. Accordingly, it generates a reply, writes it in a text file and emails it to user. Through several such interactions and correspondence emails it finally fixes a meeting and makes its entry in user's google calendar. It can also initiate a meeting request and follow the same procedure later.

4. SYSTEM ARCHITECTURE

There are four modules in the systems architecture.

i. Presentation layer
ii. Service layer
iii. Data access layer
iv. Database layer

Presentation layer: It is the interface where the user communicates with Donna. The output of this layer feeds the input to the service layer. There are two modules in this layer: the interface for Donna and a mail server (Gmail). The mail server module access the API module from the service layer.

Service layer: This layer provides services of creating and appending data to the files that are created for storing of data used for sending and receiving Email. The service layer consists of the web based services that Donna uses, like the Gmail API, the calendar API, Program O, etc.

Data access layer: This layer is the intermediate layer in the system. It carries out functions of parsing the data between the database and the front end of the system. It uses the Pattern matching algorithm and communicates with the Database layer to retrieve the matched patterns.

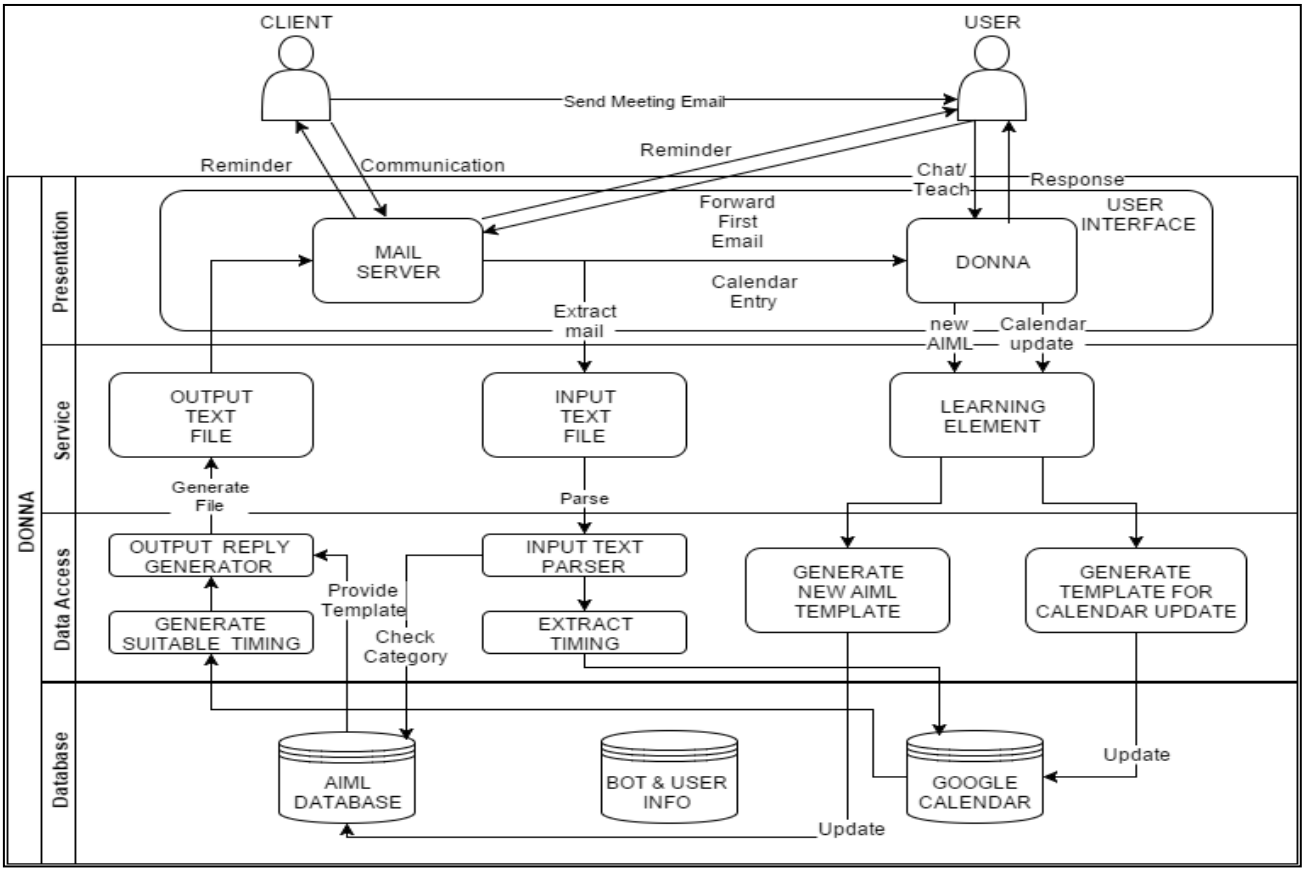


Fig 1: System Architecture

Database layer: It consists of three main databases. The first is the AIML database consisting of all the AIML files. The second comprises of the information regarding the user as well as the bot. The user's Google calendar data is stored in the third database.

5. IMPLEMETATION

For the implementation of Donna, a web server with Internet access, PHP 5+, MySQL, Xampp, Composer, and Gmail and Google Calendar APIs were used. The algorithm or the approach which has been used to develop this system can work with any other programming language or database manager.

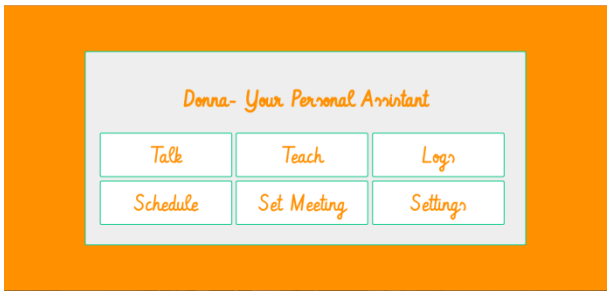


Fig 2: GUI

Following are the services which are used by Donna to connect to Gmail and calendar, to reply to emails using pattern matching (Program – O) etc.

1. Google APIs

Google APIs are application programming interfaces (APIs) developed by Google. Google APIs help us in authorization with Google Services. Google APIs includes Search, Gmail, Translate or Google Maps. Third-party applications can use the following APIs to for authorization in order to use Googles services [9].

1.1 Calendar API

One can use the Google Calendar API to find and view public calendar events. If authorized, private calendars and events on those calendars can also be accessed and modified [10]. As Donna's main function is to set up meetings, she needs to have access to user's calendar so as to check the available time slots. Also, Donna will update the calendar once a meeting is set so that the user is reminded accordingly.

1.2 Gmail API

The Gmail API is a RESTful API that can be used to access Gmail mailboxes and send email. For most web applications (including mobile apps), the Gmail API is the best choice for authorized access to a user's Gmail data. The Gmail API gives a flexible, RESTful access to the user's inbox, with a natural interface to Threads, Messages, Labels, Drafts, and History [11]. From the modern language of our choice, the application or system can use the API to add Gmail features like:

- Read messages from Gmail
- Send email messages
- Modify the labels applied to messages and threads
- Search for specific messages and threads

Thus, Donna will use this API to read the users email and also to reply to the client. The email reading and email sending functions will be used from this API.

2. Program – O

Program O is an AIML interpreter written in PHP, and uses a MySQL database to store chat-bot information, including the AIML files used to formulate the chat-bot's responses [12].

Program O supports the creation of multiple chat-bots. The basic skeleton framework for Donna is constructed using Program O. It lets a user set the bot personality, upload and integrate AIML files with the bot, connect with the database. It also has authorization feature, conversation log storing and a teaching element which can be used to teach the bot. All these features are used in Donna's web interface. Thus, the Program O library is the integral and most important part of Donna.

There are two possibilities for meeting fixing:

1. Where the user wants to initiate the meeting.
2. Where the user receives an email regarding the meeting.

5.1 User initiates meeting

This is a part where the USER wants to initiate a meeting and send an appointment request to the client. It is the simplest part of the entire working system. The user has to fill a form containing the details of Client's name, Client's email id, the date and time of meeting, and the reason for the appointment.

5.1.1 Get the receiver's details

The form on the web application of the system lets the user enter the specific information related to the meeting. Donna also checks the date and time to make sure it is a period in the future. Here intelligence is provided such that Donna checks for any holidays, birthdays, important events, etc. in the user's calendar and alerts the user if he selects one of these days for setting up a meeting. It is just a mere warning and the user can still go ahead and set up a meeting on that day if he wants to.

5.1.2 Formulate and send the email

Donna collects all the information from the form and enters it in a template. Then, the text file is converted s into an Email format with the subject set as "Regarding an Appointment". It stores the Email address of the client into a different variable and uses it to send the file to the client.

5.2 User receives meeting message

5.2.1 Pull the message from user's account.

There is an existing system Amy (x.ai), which also performs the function of setting up meetings. In that system we have to forward (CC) the email to Amy [8]. Unlike Amy, Donna does the work intelligently. Donna extracts meeting related messages from user's account by matching some keywords. Hence, the user does not have to forward the email. Once a meeting related email comes in the user's inbox, it is checked for keywords and patterns; and if it satisfies to be a meeting related email, it is directly forwarded to Donna.

5.2.2 Read the email

When Donna receives the email, it checks for the top UNREAD mails received. The Gmail API provides the functionality of checking the labels of the emails which helps to extract the unread emails. It will select the first unread email (the email that is received first), read the contents of the email and store it into a text file which is then passed to the chatbot (Program O) for parsing and reply generation.

5.2.3 Pattern matching

Pattern matching is a type of natural language processing in which the system searches for keywords in a sentence and based on that keywords, it searches the database for a match and replies with the corresponding answer. This system uses an open source software Program-O that uses a pattern matching algorithm for reading a sentence and generating a reply.

The system reads the content of the email from a text file. It will separate the E-mail ID of the person who has sent the email, which is required later to send the reply. After separating the Email-ID, the following steps are taken to parse the email.

Algorithm:

1. The system reads the text file and splits the whole data into chunk of sentences by tokenizing it.
2. Initially, it analyzes the first sentence, reads the first and the last word of the sentence and counts the number of words in the sentence.
3. It matches the AIML pattern with the first or the last words, and filters all the irrelevant pattern matches.
4. It tracks a score for the pattern and word matches i.e. whether the pattern is fully matched, common word match, default match, etc.
5. According to the scores of the pattern, it finds out if the pattern belongs to an AIML category.
6. It then initializes the sentence, matches the pattern, builds a nouns and verb list and accordingly generates a reply.
7. The procedure repeats until all the sentences are parsed and a reply has been generated.

5.2.4 Check the calendar

Donna while parsing the email, reads the condition of the meeting sent by the client. Using Google Calendar API, Donna checks the user's calendar whether he is free for the particular time slot of that day and accordingly generated a reply. The schedule of the user is saved in the database for easy access. Also, the user is able to add or modify an event in the calendar directly from the GUI.

5.2.5 Reply Generation

As discussed earlier, Donna stores the Email-ID of the client so as to send the generated reply to the corresponding Email-ID. After reading the email, the system generates a reply for a sentence which is stored in a text file. It repeats the process of parsing the sentence and generation a reply, and appends the output to the file.

5.2.6 Send the email

After the process terminates, using Gmail API and a PHP mailer library, Donna sends the text file in an Email format to the client.

5.2.7 Update calendar

As soon as an appointment is scheduled for the user, Donna immediately updates user's Google Calendar with the following meeting and blocks the slot as Meeting. Ultimately, it sends an invitation to both the parties i.e. the user as well as the person seeking an appointment with the user. Thus an event is created in the calendar.

6. EXPERIMENTAL SETUP

To perform the tests, two Gmail accounts were created; one for Donna donna.paulsen4269@gmail.com and another for Harvey harvey.spector4269@gmail.com. Two tests were conducted:

1. To check the efficiency of the system; where time taken to generate a reply for an email depending on the size was calculated for emails with varying size and contents.
2. The learning capacity of Donna; where erratic replies were detected and Donna was taught how to reply in those cases.

To conduct the first test, a number of emails were sent to Donna where each email differed in size (the unit to measure size is taken to be the number of lines) and in contents (with images, attachments etc.). The table below shows the decrease in efficiency as the size of the email increases.

Table 1. Efficiency Measure

Size of email (Number of lines)	Time taken to generate reply (in seconds 's')
1	7.50
2	8.79
5	9.90
10	14.30
15	16.00
20	17.08
50	48.00

From the observations it can be deduced that the larger the email, lesser the efficiency.

Now for the second test, consider an example. Donna was asked a question as follows:

Client: "How are you?"

Donna: "I am functioning within normal parameters."

This response clearly reveals that the person is talking to a bot (or an utterly sarcastic person). So in this case Donna was taught to pretend to be human. Following interface shows how the bot is taught.

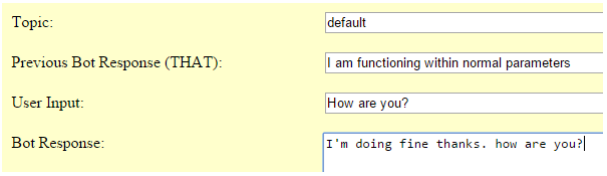
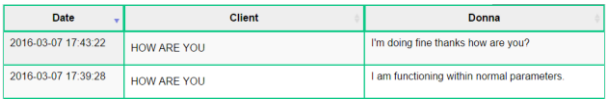


Fig 3: Teaching Interface

Thus, the ideal response that Donna should give is:

"I'm doing fine, thanks. How are you?"

Thus, after teaching Donna about this, the same question was posed again. The following image shows the reply to a question before teaching and reply to the same question after teaching.



Date	Client	Donna
2016-03-07 17:43:22	HOW ARE YOU	I'm doing fine thanks how are you?
2016-03-07 17:39:28	HOW ARE YOU	I am functioning within normal parameters.

Fig 4: Reply from Donna (before and after teaching)

It is thus evident from figure 4 that Donna is a bot capable of learning and this is the module which makes its Artificial Intelligence apparent.

An experiment was also conducted to test the quality of replies that Donna gives to the questions posed. Even though Donna is a computer Program, it does not reply perfectly to every question. In this test, replies were rated from on a scale of three, one being poor, two being average and three being excellent. Following are three examples for each type of rating.

2016-03-13 19:19:13	HELLO DONNA I AM NAMITA MHATRE	Hello there. In the context of undefined, I don't understand " DONNA I AM NAMITA MHATRE."

Fig 5: Poor Reply (Rating - 1)

2016-03-13 19:14:29	HELLO DONNA I AM NAMITA MHATRE	Hello there. I like the way you talk.
2016-03-13 19:14:29	HOW ARE YOU	Fair to partly cloudy.

Fig 6: Average Replies (Rating - 2)

2016-03-04 15:34:42	CAN I GET AN APPOINTMENT WITH HARVEY SPECTOR	Yes, HARVEY SPECTOR is free tomorrow at 2pm.

Fig 7: Excellent Replies (Rating - 3)

The experiment was tested on a log of about 700 messages approximately for the period of six months and it was noted that Donna replied poorly for about 5% of the messages, average for about 60% of the messages and excellently for about 35% of the whole messages. These statistics are close to those of other existing chat-bots. The below diagram represents the graphical view of the statistics.

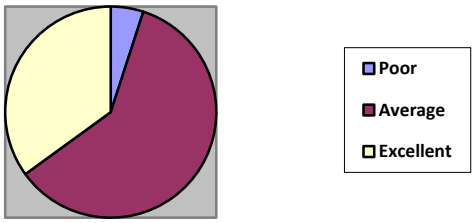


Fig 8: Performance Statistics

7. RESULTS

From the first experiment, it can be concluded that efficiency decreases with the increase in the size of emails. Generally, professional communication consists of terse emails without much superfluous talk. Thus it is assumed that the efficiency of the system won't be wounded much as this application is for managing schedule and setting up meetings. After studying some professional emails (regarding meeting) it can be said that their size usually rounds up to no more than 30 sentences. Thus, an efficiency of 30 seconds is good for the system as compared to a human assistant who usually takes a lot more time than that to reply to emails.

The second experiment proves that Donna is an interactive chat-bot consisting of a learning element. Thus, whenever the user finds a particular reply to be incorrect, he may teach

Donna the correct reply for the same question and she learns. This is very useful for this particular application as each user might want to have different answers for questions depending on personal choices. For example, when the person is not available for a meeting, he might not want his secretary to reveal the reason why he is not available in that particular time slot. But as information about his schedule is available to Donna, she might disclose it to a person asking to schedule a meeting for the same time slot. At such times, the user could teach Donna what to talk about and what not to talk about.

As shown in the third experiment that Donna replies averagely about 60% of the times. Humans are bound to make mistakes many times. They misunderstand a statement or a question and may reply irrelevantly. Donna is a piece of artificial intelligence but making some errors makes the system indistinguishable from a human assistant.

8. CONCLUSION AND FUTURE WORK

Using pattern matching algorithm, a system that can act as a virtual personal assistant to plan user's work and schedule his meetings was successfully designed.

In terms of the efficiency of the system to respond within a stipulated time period, which achieved overall 70% efficiency, it can be concluded that the system is capable enough to be implemented in the practical world.

Furthermore, the system can be enhanced by including various modules. An important module that can be incorporated is setting the priority of client with whom the user should set meeting, in case of clash of request by multiple clients. This could either be done by getting information about user's personal relationship with the client through machine learning tools and giving priority to close friends and frequent colleagues, or by simply asking priority of client to the user.

One of the major part in the system can be enriched is adding the functionality of scheduling meeting with multiple clients for race condition like same time of meeting. Also, the location factor has to be included so as to allocate a meeting place appropriate for both the user as well as the client.

9. LIMITATIONS

From the results, it is evident that as the number of lines in the email increases, the efficiency of the system decreases to a great extent. So as a limitation to this system, if the client sends a huge chunk of data in a single email, the system might hold back and take a long time to respond to the request of the client.

Also, Donna is not able to handle conversations related to group (conference) meetings. If multiple clients approaches Donna for the same conference meet individually, the system would allocate different timings for each client. Moreover, the chat-bot is not considering location of the meeting. If the user is busy with a client and free for another slot, Donna might set up an appointment right after the current meet say at a distance of 20 Km. The user might take time and end up reaching late for the meeting.

10. ACKNOWLEDGMENTS

Especial thanks to Ms. Swati Mali for her guidance throughout the project and Ms. Nirmala Shinde for her insightful comments.

11. REFERENCES

[1] ChatbotWikipedia,"https://en.wikipedia.org/wiki/Chatbot", last retrieved on 13-03-2016.

[2] ChatterbotWikipedia,"https://en.wikipedia.org/wiki/Chatterbot", last retrieved on 13-03-2016.

[3] Supratip Ghose & Jagat Joyti Barua, "Toward the implementation of a Topic specific Dialogue based Natural Language Chatbot as an Undergraduate Advisor", 2013, 978-1-4799-0400-6.

[4] Emanuela Haller, Traian Rebedea, "Designing a Chat-bot that Simulates an Historical Figure", 19th International Conference on Control Systems and Computer Science, 2013, 978-0-7695-4980-4

[5] "Shallow Parsing Natural Language Processing Implementation for Intelligent Automatic Customer Service System", ICACSIS, 2014, 978-1-4799-8075-8

[6] Salto Martínez Rodrigo & Jacques García Fausto Abrahaman, "Development and Implementation of a Chat Bot in a Social Network", Ninth International Conference on Information Technology- New Generations, 2012, 978-0-7695-4654-4

[7] A.L.I.C.E. Artificial Intelligence Foundation, "https://alice.pandorabots.com", last retrieved on 12-02-2016.

[8] X.ai, "https://x.ai", last retrieved on 20-02-2016.

[9] GoogleAPIsWikipedia,"https://en.wikipedia.org/wiki/Google_APIs", last retrieved on 07-03-2016.

[10] GoogleCalendarWikipedia,"https://en.wikipedia.org/wiki/Google_Calendar", last retrieved on 07-03-2016.

[11] Gmail API wiki - Stack Overflow, "http://stackoverflow.com/tags/gmail-api/info", last retrieved on 07-03-2016.

[12] GitHub - Program-O, "https://github.com/Program-O/Program-O", last retrieved on 07-03-2016.