

Selena: A Voice Enabled Intelligent Clinical Assistant

Mandar S. Deshpande¹, Sagar R. Bhave¹, Ashish Kumar R. Bidap¹, Tushar S. Gote¹, Prof. Parag S. Kulkarni¹

NBN Sinhgad School of Engineering, Ambegaon (Bk), Pune¹

Abstract: This paper illustrates the design of an end to end application for an intelligent personal assistant, named Selena, who accepts the queries in the form of voice or text and responds same with natural language processing. Selena has been modelled as an alternative for the physical presence of a medical assistant at a hospital or a clinic. Selena would develop an interactive chat session with the patient enquiring about the various details and patient history and later generates a formatted document which holds the data extracted from that chat session and which would be further passed on to the doctor. Selena withholds well established open infrastructure for speech recognition, speech synthesis, machine learning and knowledge discovery from data.

Keywords: Machine learning, Chatbot, Pattern Recognition, Speech Recognition, Speech Synthesis, Java Speech API, Knowledge Discovery, iText Pdf Generator, Selena: A Clinical Assistant.

I. INTRODUCTION

The main idea behind the implementation of Selena: A Clinical Assistant, came into picture from the A.L.I.C.E. Chatbot by Dr. Richard Wallace of Pandorabots. From this AIML based Chatbot, user requests for an appropriate query and bot gives logical and sensible response according to the asked query. Targeting the audiences to interact freely with the Chatbot, is to introduce a voice input over the normal text input and a better user interface to make the conversation more interactive, are the main things to look after while developing Selena. Keeping the Chatbot as simple at front end and developing the complicated bot brain at back end is the most crucial part in its design. As Java is ready to provide its all functionalities to develop and integrate the full-fledged modules for Selena, we kept Java as a main programming language for its implementation. Thus, taking it into consideration, we developed an application, Selena, which is a Clinical Assistant which acts as an alternative to the medical assistant at hospitals and clinics.

Who is Clinical Assistant? Most clinical medical assistants handle the administrative work for a doctor's office or health clinic. This may include scheduling appointments, directing phone calls, filing documents and getting patients where they need to be.

Related Works:-

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

Artificial Linguistic Internet Computer Entity of ALICE AI Foundation by Dr. Richard Wallace of Pandorabots, is a text based conversational agent.

2. ELIZA

A classic A. I. Program named as ELIZA psychiatrist is a simple personal pronoun reversal program, to create the illusion of understanding when in fact it had none. The idea was to turn around and "reflect back" anything the client said, by replacing first person pronouns ("I" and "me") with second person pronouns ("You"). (Pandorabots.com, 2005).

3. TeenChat

A text based Chatterbot System for sensing and releasing adolescence stress.

4. ViDi

Virtual Diabetes (ViDi) physician which is a Chatbot for diabetes education activity using One Match and All Match Categories Algorithm for Keywords matching in Chatbot.

II. IMPLEMENTATION

Keeping java as a main programming language, we inherited its most important feature i.e. 'Maximum Cohesion and Minimum Coupling', meaning, each independent module (as described in system diagram) should work successfully on its own and while integrating those separate modules, one whole Chatbot, runs charmingly. Selena is created with following modules from the diagram and the procedure carried out is discussed later.

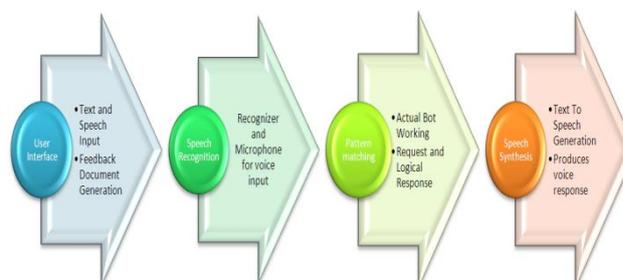


Figure 1: System Diagram

The above modules form one complete Chatbot design a voice enabled application Selena: A Clinical Assistant. First of all Java Interpreter acts as the base since it is responsible for the request-response processing and the algorithm to match an appropriate pattern, as well as the document generation. The brain file built for Selena is a repository of AIML and AIMLIF files which helps in

efficient pattern matching paradigms. Java Speech API is used for the text to speech and speech to text conversions. The following section describes the actual working of Selena's modules.

1. User Interface:

To achieve excellent user experience, user interface must be kept as user friendly as possible and Java Frames provides whatever a normal user wants from any application. A microphone button performs dual tasks, accepting a text input typed into question box or voice input from the device microphone. Following figure shows the typical user interface for Selena: A Clinical Assistant.



Figure 2: Front End of Selena: A Clinical Assistant

Question box near the mic button, accepts the text input and the answer box at the top produces the response generated from 'Pattern Matching' module. At first place, 'User Interface' module initializes the main parts of the Chatbot, allocates the recognizer and microphone from the speech recognition module, and starts the Pdf Document to write the conversation history. Once, the call is forwarded to the Pattern Matching Module, the actual working of Chatbot, starts there. User has two options to give its input either by text format or by speech. If user directly presses the mic button on GUI, Selena waits for the speech input to accept and then transfers the call to the pattern matching module. Finally, Answer matched from the Selena's brain is then displayed along with the voice into the answer box. At last, on clicking the 'Exit' button, application generates the well formatted document in Pdf form for printed conversation and successfully exits the Chatbot.

2. Speech Recognition System:

Java Speech API along with Sphinx-4 API provides the functionality to accept the speech input from device microphone. Recognizer and the Microphone are the main objects to enable the speech acceptance from the microphone. Actual recognized word is stored as a final result and this same result is passed to the 'Pattern Matching' module as a request from the user to match an appropriate response. A separate dictionary file is used to configure the probable words spoken by user and to be recognized by Selena. This dictionary file is in the JSGF (Java Speech Grammar Format) and can be accessed via a XML File. XML File acts as a word recognizer, linguistic, acoustic and grammar configuration manager. Sphinx Loader sets the default voice frequency at 130Hz to 6800Hz for better recognition of voice.

3. AIML and Brain of Selena:

AIML (Artificial Intelligence Markup Language), a XML based scripting language is a heart of Selena and the AIMLIF (Spreadsheet like CSV files) is a brain of Selena. In AIML files, actual Patterns (questions asked by user) and Templates (answers generated by Chatbot) are stored to be matched by the Graphmaster Algorithm with the help of NodeMapper method. Whereas, AIMLIF files contain the six different fields separated by commas and to be matched with their linked AIML Files. The main part of machine learning and artificial intelligence in Selena is situated within these AIML files where, tags like <that>, <think>, <topic>, <person> and <srail> are used to train the Chatbot. The actual brain of Selena, contains the questionnaires to accept the details and medical history of a patient. A normal Chatbot gives answers to the user asked questions but, Selena seeks for the answer from patient to the question asked by herself.

The Pattern Matching algorithm along with other functionalities of the pattern matching framework, follows a specific path from instantiating a chat session, loading AIML Files, reading Sets and Maps Filesto matching with the contents of the AIMLIF Files.

Following code snippet shows the Machine learning and Pattern Matching of 'patterns' to 'templates' of AIML and AIMLIF files in Selena.

```
a) selena.aiml:-
<aiml>
<topic name="greet">
<category>
<pattern> My Name is * </pattern>
<template>
Hello<star/>Glad to Meet You! What is Your
Age?<think><set
name="Patient_Name"><star/></set></think>
</template></category></topic><category>
<pattern>My Age is *<pattern>
<template>
<get name="Patient_Name"/>you are <star/> years
old!</template>
</category></aiml>
```

```
b) selena.aiml.csv:-
0,My Name is *,*,*,Hello<star/> Glad to Meet You! What
is Your Age? <think><set name
="Patient_Name"><star/></set></think>,selena.aiml
0,My Age is *,greet,*,<get name="Patient_Name"/> You
are <star/> years old!,selena.aiml
```

c) Its Output will be:
Patient: My Name is Steve
Selena: Hello Steve Glad to Meet You! What is Your Age?
Patient: My Age is 22
Selena: Steve, you are 22 years old!

- In above AIML File (selena.aiml),
- i. <think>tag will save the patient name without knowing to user hence, can be used to set the patient's name for next question.
 - ii. <topic>pattern will save the topic name to proceed the conversation with prior context.

iii. <star>pattern is used to match a * closure containing anything in it.

And AIMLIF File (selena.aiml.csv) contains 6 fields and they are:

'activation count', 'input pattern', 'that pattern', 'topic pattern', 'template', 'filename.aiml'

Following diagram displays the exact working of Pattern Matching from Selena's brain.

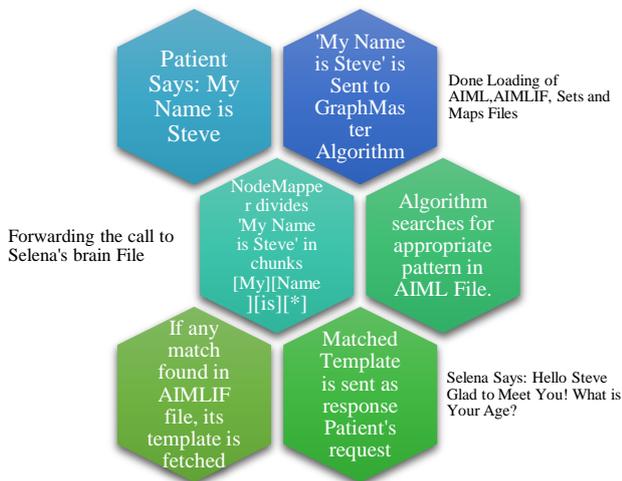


Figure 3: Working of Pattern Matching Module

4. Speech Synthesis:

To generate a voice output, Java Speech API in combination with FreeTTS and Mbrola Female US English voice is the best solution. As we decided to make a Female Chatbot, we used female English voice otherwise, normal speech synthesizer generates basic robotic male voice. Speech synthesis module initializes the 'Voice Manager' and 'Voice' to speak out the text parameter by setting a system property for Mbrola Female US English voice. The following command shows the output generated by Speech Synthesis Module.

```
voice.speak("Hello, Glad to Meet You! What is your Name?");
```

5. Knowledge Discovery from Data:

At the end of a conversation, when patient's appointment is fixed with doctor on upcoming date, Selena generates a well-structured Pdf Document with only required amount of data. For example, if patient says, 'My Name is Steve' then on the Pdf Document, only 'Name: Steve' is going to be written, other unwanted words are simply deprecated. Likewise, whole sheet describes only patient's details and one's medical history, none other unnecessary data. This document starts recording the patient's details at the very beginning of the conversation and successfully closes the document with the click on 'Exit' button. This facility can be introduced by using iText Pdf Generator which creates and edits the Pdf document using Java File Handling Functionalities.

III. PROCEDURE

- Firstly brain file is created for Selena with respect to clinical related queries.

- After Loading of the brain file, Selena waits for the patient requests.
- Patient need to start the conversation in the form of text or voice input.
- Selena receives the request and forwards to Pattern Matching Module
- Pattern Matching Algorithm is applied which matches the pattern to the appropriate template from the brain of Selena. This matched template is then spoken aloud as well as displayed in the form of text.
- Additionally knowledge discovery from the requests and responses is done to generate a document after processing. This document would be a well formatted one which would help the doctor for any further interventions.

IV. RESULT AND ANALYSIS

On testing, Selena: A Clinical Assistant, performed very much up to the mark with the text inputs, results were exactly accurate and efficient. In case of the voice input, Selena was very much reliable but sometimes due to accent mismatch voice input gave a close to appropriate answer. Furthermore, the document which is generated from the text or voice input is perfectly mined and only contains data which is required. Following figure showcases the first entry of Selena which is 'greet' to invoke a conversation.



V. FUTURE SCOPE

1. Implement a Language Independent System so that the patient who is very much comfortable with other languages than just English.
2. Accent Free speech Recognition System such that system would be more accurate and specific.
3. Giving the Home Remedies at the primary level if Patient's disease is not so severe.

VI. ADVANTAGES

1. Develops a comfort level for the patient.
2. Avoids the gender differences situations where the patient cannot be lucid discussing certain issues.
3. Interactive conversation of Selena binds the user in order to respond to the clinical based queries.

VII. CONCLUSION

This work introduces Selena: A Clinical Assistant, an application which proves to be useful in order to successfully act as a replacement for the physical presence of the medical assistant at the clinics or hospitals. We show that with the help of Selena, human efforts can be reduced to far more extent as well as the patient would be very much comfortable interacting with Selena avoiding the gender differences. Additionally knowledge discovery from the data and processing it on a document helps to avoid any human intervention resulting in error free environment.

REFERENCES

1. Setting Up FreeTTS:-<https://community.oracle.com/thread/1273833?start=0&tstart=0>
2. Mbrola Voices:- <http://tcts.fpms.ac.be/synthesis/mbrola.html>
3. Sphinx-4 API:-
<http://cmusphinx.sourceforge.net/wiki/tutorialphinx4>
4. (alice.pandorabots.com,n.d.): <http://alice.pandorabots.com/>
5. (Pandorabots.com, 2005): Be your own Botmaster by Pandorabot.com, A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) AI Foundation.
6. (Parag Kulkarni, 2015): Artificial Intelligence Building Intelligent Systems by Parag Kulkarni and Prachi Joshi.
7. (Singh, 2015): AIML Based Voice Enabled Artificial Intelligent Chatterbot by Imran Ahmed and Shikha Singh.
8. (Vinay K. Chaudhri, 2014): A Case Study in Engineering a Knowledge Base for an Intelligent Personal Assistant