ChatBot for University

Saranya Soundar Rajan
Arshiya Hayatkhan Pathan
Zeeshan Ali
Vasu Bansal

CMPE 272- Enterprise Software Platforms
San Jose State University, California

Abstract— **Universities have a plethora of information and it's a challenge to get this across to a prospective student on demand basis. For any Undergraduate or Graduate Programs, the applicant knows the process of researching university programs is a time-consuming and complicated process. An enormous amount of research is required to find the information they need to get into the right university. As students have many queries regarding university as well as course requirements, the applicant has to contact various individuals and departments and spend lot of time in browsing the internet to look for information they need. They have to go through lot of pages to find out information for financial aid, courses, programs, credits. To get admission to graduate school, applicant need to apply through university's online application form and submit supplementary documents and proofs. For Prospective students, it is important to complete their application within deadline. When the student cannot find the information in FAQ or they don't know where to look for, they end up emailing graduate advisors and wait for a response. This poses a real problem because it can have students waiting on one end for an answer. With emerging technologies, universities have a way to solve this problem using Chatbots. Chatbot understands what the student is saying (natural language understanding) and replies to the student in a logical way (natural language processing). There are some scenarios where chatbot can advise a student on appropriate courses, assist a student in filling up an application form, assist a new student on Campus.**

Keywords – Chatbot, bot, Natural Language Processing, Artificial Intelligence, Text-to-Speech, Speech-to-Text.

I.INTRODUCTION

A recent study says, after the innovation of mobile technologies, digital messaging becomes the preferred method of communication for younger generations. Universities receive phone calls or emails as the primary source of contact with potential students [8]. With huge volume of emails, the responses to the students might be delayed. By implementing Chatbots in university, the student's queries will be answered instantly.

Chatbot conducts conversation via auditory or textual methods (Text-to-Speech or Speech-to-Text). A formal definition of chatbot is "chat robots" which provide highly engaging, conversational experiences, through voice and text, that can be customized and used on mobile devices, web browsers, and on popular chat platforms [10]. Chatbots itself does not have any intelligence. To gain intelligence, chatbots are programmed and trained with artificial intelligence or machine learning algorithms to provide a service to individuals/business that interact with it via instant messaging [8]. It is like any other social networking platforms – Facebook, Twitter or Slack where users can communicate with other users. Chatbots can be used in businesses for solving customer related queries or for an individual user, trained to do tasks ranging from getting weather report, order a product online or google for an information. Chabot's intelligence will be improved with constant training with situations. It works similar to Apple's Siri.

The process of building a chatbot can be categorized into two parts: one understanding the user's intent and second is to respond accordingly. Natural language processing and machine learning capabilities of Dialogflow (previously api.ai) creates the conversational agent. A Dialogflow agent represents the conversational interface of bot. Training chatbot using dialogflow involves identifying entities and intents. An agent is essentially the container which includes intents, entities, and the responses that are to be delivered to user. Intents are the mechanisms that pick up what user is seeking (using entities) and guide the agent to respond accordingly. The DialogFlow agent needs to know what information is useful for answering the user's request. These pieces of data are called entities [1].

II. CHATBOT ARCHITECTURE

Chatbot Architecture with response selection

Chatbot with response selection has been used for this Project. This is based on Retrieval-based models which uses a repository of predefined responses. Based on the user's queries, it picks an appropriate response based on context. It is a rule-based expression where systems do not create new text or sentence, they just pick a response from the predefined responses whereas Generative models create new responses using Machine-Learning Techniques. It requires lot of training and data to be trained with.

The basic architecture of Chatbot application is shown in Fig.1

Bot responds to user's query using different sentences or words. The chatbot have a separate response generation and response selection modules [9]. Message processing understands the user's query by looking through the intents and entities in Natural Language Processing API such as Microsoft bot framework, Wit.ai, API.ai, and Aspect CXP-NLU.

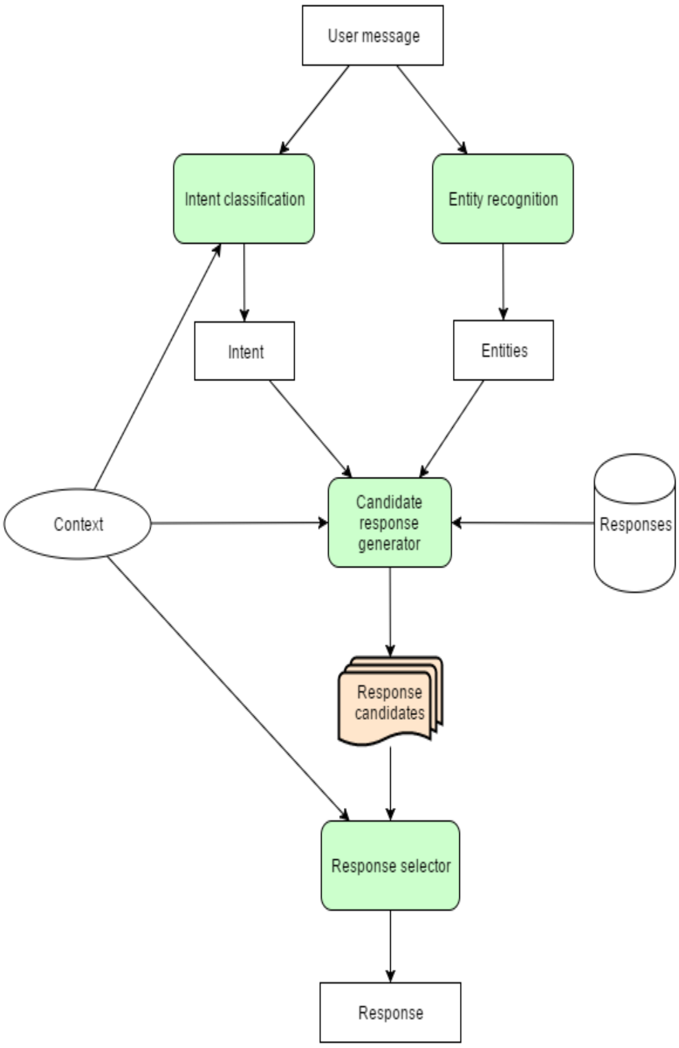


Fig 1: Chatbot Architecture

Intent classification module identifies the intent by looking for certain words in the user message. Then it selects one of the predefined intents, though more sophisticated/trained bots can identify multiple intents from a single message. Intent classification can use context information, such as intents of previous messages, user profile, and preferences. Entity recognition module extracts structured bits of information from the message.

III. System Architecture And Flow

The client posts a question to the bot (Voice or Text based queries). Web Speech Recognition API is used to listen and convert the client's voice to text. The text will be sent to Node.js server using Socket.io. Socket.io helps communicate this to the server. Express, a Node.js web application server framework is used to run the server locally.

The architecture for University based Chatbot is shown in Fig.2

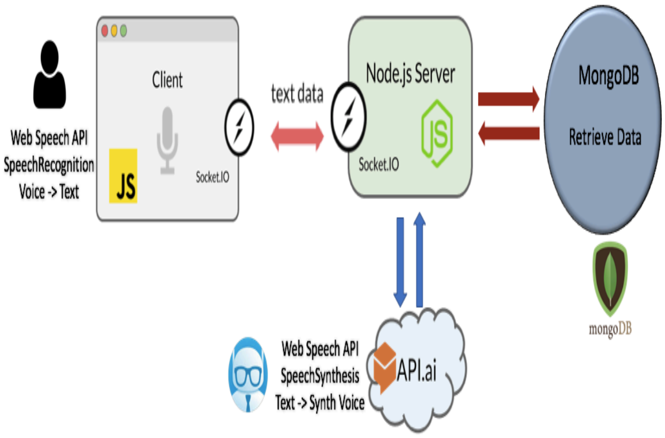


Fig 2: System Architecture

[7] To enable real-time bidirectional communication between the server and the browser, Socket.IO have been used as it avoids frequent loading of the page or to send AJAX requests at regular interval. At the server, the question is posted to API.ai and Text message is sent to Natural Language API (DialogFlow) as text string. Using its natural language and text to intent processing, the questions are scanned for intents. Each intent is further mapped to a specific answer. Once the response is loaded, Speech Synthesis converts the Text to Synth Voice and the answers are further returned to the node.js server. At the server, data for current students is stored in MongoDB and appended to API.ai's outputs if required. The output is then communicated to the client using socket.io. The system will give the text output and while giving it, it will give a voice output as well. To retrieve the information of current students, MongoDB has been used. Chatbot can be deployed on web as PaaS (Platform as a Service).

IV. Technology Overview

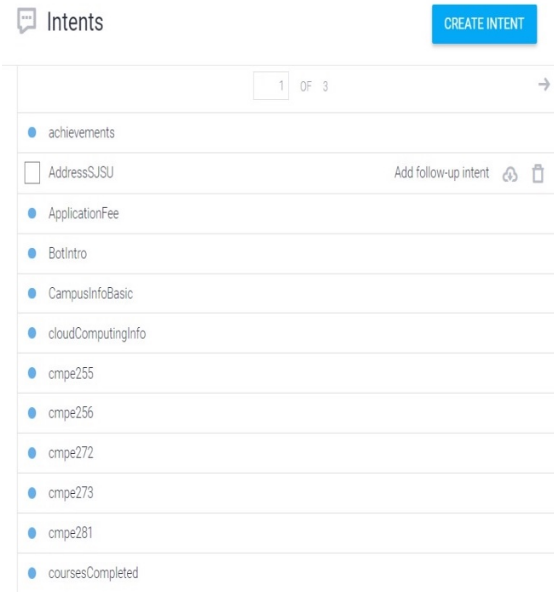
Restful API refers to an API that uses HTTP requests to GET, POST, PUT and DELETE data. [2] We have used concepts of REST API for this application primarily due to its clear separation between client and the server as well as visibility, reliability and scalability offered by REST [2]. The system consists of a node.js server on the backend. It provides for an event based, asynchronous I/O framework [3]. It uses Express as the server framework for Node.js. It helps us organize the application into an MVC architecture on the server side [3]. It uses Bootstrap and JavaScript at the front end(client). The application also uses Socket.io, a JavaScript library that enables real time bidirectional communication between server and client(s). [4] API.ai is the natural language processing based interface that we have used in the application using the API.AI Nodejs SDK. [1] It helps us build and test conversational scenarios by creating intents as shown in Fig 4. We have used MongoDB is the database primarily due to the flexibility and low latency offered by it. For site administrator and current students, login authentication and session management has been implemented using the node-passport.js npm package. This application is deployed as SJSUbot on Heroku.

API KEYS (V1)

Client access token

Developer access token

Fig 3: API Keys



- achievements
- AddressSJSU
- ApplicationFee
- BotIntro
- CampusInfoBasic
- cloudComputingInfo
- cmpe255
- cmpe256
- cmpe272
- cmpe273
- cmpe281
- coursesCompleted

Fig 4: DialogFlow Intents

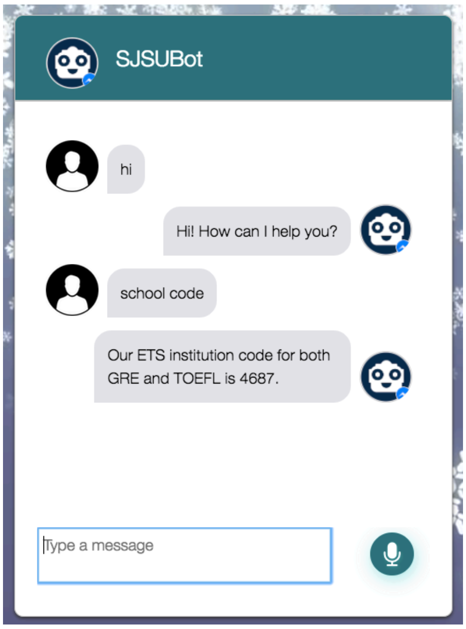
V. Functionalities

A. *Prospective Student Functionalities*

The student will be directed to Bot from the Home Page. Like any other messaging Platform, the student/user uses the input controls - microphone or textbox to talk to the bot as shown in Fig 5.

Fig 5: Input Controls

As explained in system flow, the user's question is sent to our server via Socket.io, our server does the processing and return the result back to the client. The listener gets this message, then passes it to the function which injects it into the DOM as shown in Fig 6. Text-to-Speech Engine converts the text from the server and plays it back through the connected hardware, aka speakers.



hi

Hi! How can I help you?

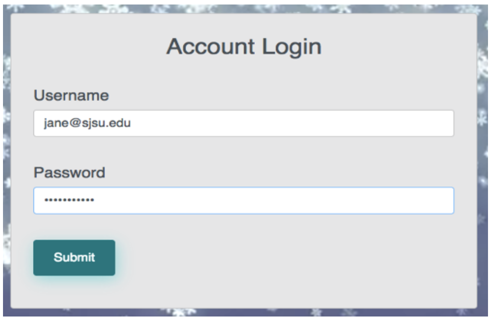
school code

Our ETS institution code for both GRE and TOEFL is 4687.

Fig 6: Prospective students Botpage

B. Current Student Functionalities

The currently enrolled students will be directed to a Login Page from the Home Page and will be asked to login using their login credentials as shown in Fig 7 & 8. In addition to getting answers for generic queries, the current students can also use the bot for their academics and semester related queries pertaining to them, specifically. Students get to enjoy the far better than personalized experience rather using a portal to navigate through several pages to get an information.



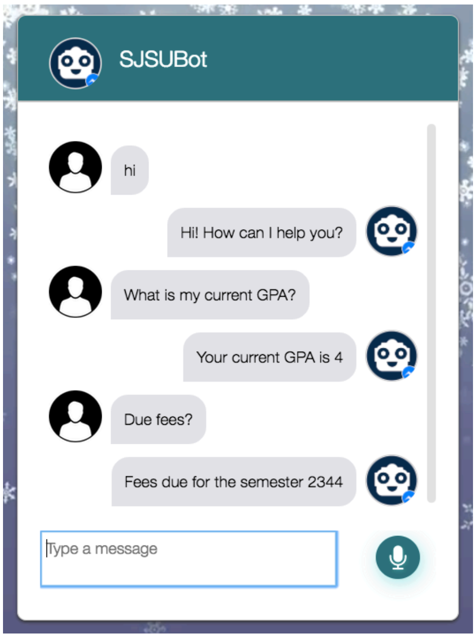
Account Login

Username: jane@sjsu.edu

Password: •••••••••••

Submit

Fig 7: Current student login



SJSUBot

hi

Hi! How can I help you?

What is my current GPA?

Your current GPA is 4

Due fees?

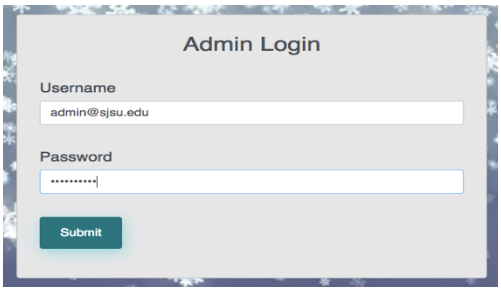
Fees due for the semester 2344

Type a message

Fig 8: Current students Botpage

C. Administrator Functionalities

Student's information can be added to the database by the admin alone by securely logging into the student registration page. Admin Login and Registration are shown in Fig 9 & 10.



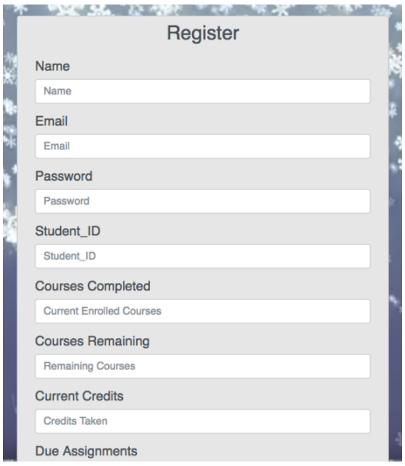
Admin Login

Username: admin@sjsu.edu

Password: ••••••••••

Submit

Fig 9: Secure admin login



Register

Name: Name

Email: Email

Password: Password

Student_ID: Student_ID

Courses Completed: Current Enrolled Courses

Courses Remaining: Remaining Courses

Current Credits: Credits Taken

Due Assignments

Fig 10: Student Registration by admin

VI. Functional Requirements

- 24-hour student accessible
- Graphical Interactive System
- Keep track of progress of current student
- Correctively Identify abstruse queries [4]
- Ease Application Process for students

VII. Non-Functional Requirements

- High performance measurement
- High availability
- Scalability to modify the data taken by the application based on the current student database
- Fault tolerance so that the system is stable even if few functionalities fail
- Maintainability so that the bot can be used to answer any number of queries [4]
- Timely Response
- Modular

VIII. Conclusion

With improved communication and the ability to offer schools major financial savings, chatbots may soon be the future of how universities communicate with students. Using the artificially intelligent Chatbot, it will serve aspiring students with the instant answers to all the queries related to application process and also for the current students with the quick reply for checking grades and due assignments. Continuously training the bot with more questions can help to make the bot reply efficiently.

IX. Future work

In this Project, we have implemented Short-Text Conversations where the goal is to create a single response to a single input. By applying advanced Machine-Learning Techniques, like the Generative model, the Chatbot will become smarter and it can plausibly engage the user in certain conversation. This technique has a feedback loop which helps the bot to learn new things when engaging with the user.

Acknowledgment

We thank Prof. Rakesh Ranjan for his guidance for completion of this project successfully.

References

[1] "Dialogflow", *Dialogflow*, 2017. [Online]. Available: https://dialogflow.com/. [Accessed: 09- Dec- 2017].

[2] "RESTful API", 2017. [Online]. Available: http://searchcloudstorage.techtarget.com/definition/RESTful-API/. [Accessed: 10- Dec- 2017].

[3] "Node.js", 2017. [Online]. Available: https://app.pluralsight.com/player?course=node.js-getting-started/. [Accessed: 8-Nov- 2017]

[4] "Chatbot Evaluation and Database Expansion via Crowdsourcing - Semantic Scholar", *Semanticscholar.org*, 2017. [Online]. Available: https://www.semanticscholar.org/paper/Chatbot-Evaluation-and-Database-Expansion-via-Crow-Yu-Xu/a950b42d3e46f4e1fb0e7e6959bc85a66246adb8. [Accessed: 10- Dec- 2017].

[5] “College Enquiry Chat Bot” Ijritcc.org, 2017. [Online]. Available: http://www.ijritcc.org/download/browse/Volume_5_Issues/March_17_Volume_5_Issue_3/1492062497_13-04-2017.pdf. [Accessed: 10- Dec- 2017].

[6] "Chappie - A Semi-automatic Intelligent Chatbot", 2017. [Online]. Available: https://www.cse.iitb.ac.in/~bibek/WriteUP_2016.pdf. [Accessed: 10- Dec- 2017].

[7] "Building A Simple AI Chatbot With Web Speech API And Node.js", *Smashing Magazine*, 2017. [Online]. Available: https://www.smashingmagazine.com/2017/08/ai-chatbot-web-speech-api-node-js/. [Accessed: 10- Dec- 2017].

[8] T. Putz, "Higher Ed: Why Chatbots are the Future of Communication", *Higher Ed Marketing Journal*, 2017. [Online]. Available: http://circaedu.com/hemj/higher-education-marketing-why-chatbots-are-the-future-of-communication/. [Accessed: 10- Dec- 2017].

[9] "Chatbot Architecture – Pavel Surmenok – Medium", *Medium*, 2017. [Online]. Available: https://medium.com/@surmenok/chatbot-architecture-496f5bf820ed. [Accessed: 10- Dec- 2017].

[10] "Chatbot", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Chatbot. [Accessed: 09- Dec- 2017].