

## Erasmus – AI Chatbot

**Jash Thakkar<sup>1\*</sup>, Purva Raut<sup>2</sup>, Yash Doshi<sup>3</sup>, Krishma Parekh<sup>4</sup>**

<sup>1,2,3,4</sup>Dept. of Information Technology, Dwarkadas J. Sanghvi College of Engineering, University of Mumbai, Mumbai, India

\*Corresponding Author: [jthakkar2@gmail.com](mailto:jthakkar2@gmail.com), Tel.: +91-98332 25265

Available online at: [www.ijcseonline.org](http://www.ijcseonline.org)

Accepted: 20/Oct/2018, Published: 31/Oct/2018

**Abstract**— ERASMUS is a chat bot on Facebook which is used to answer the queries related to the college information in one go. There has been an increasing demand for Chatbots on the websites of colleges or Educational Institutes. The college websites are generally unmaintained and filled with redundant information. Finding the required information on the website becomes a tedious task for students, teachers, parents or other users. The problem also gains a boost during time of admission, as the requirement for information increases during that time. A chatbot on the college website will help the users find the required information in a few clicks and questions. The chat bot detects the intent of the queries received from the users and accordingly parses the website to find the related information. There is no specific format the user must follow. The system uses built in artificial intelligence to answer the queries. The user can query any college related activities through the system. The user does not have to personally go to the college website for any enquiry.

**Keywords**— Chatbot, Intent, API.AI, AI, Natural Language Processing, Machine Learning, Artificial Intelligence

### I. INTRODUCTION

Querying to the various college websites especially during the admission and exams is a time-consuming job for most of the students. Reason being the websites go very slow as the number of hits increase to a very high number. Also, many students are unaware of the information available on the website and hence they end up wasting a lot of time and energy in surfing through many un-useful web pages.

The AI Chatbot is built with mere motivation of reducing the search time for students and make the websites more friendly and informative which is very important in those stressful times. We developed this chatbot using artificial intelligence algorithms that analyses user's queries and understand the user's messages. Students must query through the bot which is used for chatting and the interface is same as any other chat box. The system interprets the question and analyses it further to generate answer for the user. The system answers the queries as if it was answered by a real person. The system replies using an effective graphical user interface which implies that a real person is talking to the user.

The user should register himself and should login to the system. After logging in, the user can access various helping pages. The helping pages has the bot through which the user can chat by asking queries related to college activities. The system replies to the user with the help of effective graphical user interface. The user can query about the college related activities through online with the help of this web

application. The user can query college related activities such as date and timing of annual day, sports day, and other cultural activities. This system helps the student to be updated about the college activities.

This paper is organized in the following manner. Section I starts with the need of chatbots and its demand on college websites. Section II contain the related work on chatbots and some real-world examples of chatbots along with their popularity and pitfalls. Section III contains the proposed architecture and the technologies used to develop the chatbot. Section IV contain the steps of implementation. Section V provides us with an appropriate conclusion of the project Section VI includes the references used in the research of the project.

### II. RELATED WORK

#### A. CLEVERBOT

Cleverbot is a web application that uses an artificial intelligence algorithm to have conversations with humans. Besides the web application, Cleverbot is also available as an iOS, Android, and Windows Phone app.

Cleverbot's responses are not programmed. Instead, it "learns" from human input; Humans type into the box below the Cleverbot logo and the system finds all keywords or an exact phrase matching the input. After searching through its saved conversations, it responds to the input by finding how a human responded to that input when it was asked, in part or in full, by Cleverbot.

Cleverscript has more AI features than any other bot creation tool. Cleverscript excels in its deeper understanding,

its sensitivity to context, and its speed of development through access to millions of rows of background chat from billions of interactions recorded by Cleverbot. You can easily create your own bots – all you need is a spread sheet.



Figure 1: Popularity statistics of Cleverbot

**Pitfalls:**

To be fully functional, retailers need fast, robust connectivity that integrates these digital agents into existing infrastructure. A chatbot’s effectiveness will be limited by the retailer’s ability to give it a 360-degree view of the customer. Retailers need to combat the notion that chatbots can become potential agents of consumer fraud. Although consumers crave instantaneous and customized experiences, they will be wary of new technology if they don’t feel it is secure.

**B. ALICE CHATBOT**

A.L.I.C.E (Artificial Linguistic Internet Computer Entity) is an award winning open source natural language artificial intelligence chat robot which utilizes AIML (Artificial Intelligence Mark-up Language) to form responses to queries. The main knowledge bases of ALICE are stored on different AIML files; ALICE is available to the public for free under the GNU license. The ALICE software implements AIML (Artificial Intelligence Markup Language) a non-standard evolving markup language for creating chat robots. The primary design feature of AIML is minimalism. Compared with other chat robot languages, AIML is perhaps the simplest. The basic unit of knowledge in AIML is called a category. Each category consists of an input or question, and output or answer, and an optional context. The question is called the pattern. The answer or response is called the template. The study mainly focused on ALICE used solely as conversational agent, the study produced unexpected result as majority of the users did not like the response of the chatterbot and left bad comments on the system, the study also showed most users only interacted with the system for very short amount of time.

Table shows the statistics of data collected from the survey when students interacted with the FAQ(limited) chatterbox.

Context of conversation	Satisfactory	Unsatisfactory
Admission Info	70%	30%
Course Info	80%	20%
Faculty Info	60%	40%

Table shows the statistics of data collected from the survey when students interacted with the FAQ(full) chatterbox.

Context of conversation	Satisfactory	Unsatisfactory
Admission Info	60%	40%
Course Info	60%	40%
Faculty Info	50%	50%

**III. PROPOSED ARCHITECTURE**

An Institution’s website maybe unmaintained and filled with redundant information. This makes it difficult for students, teachers, parents or other users to find the required information. The problem increases during time of admission, when there’s haste and busy college contact lines. A college website chat bot will help the users find the required information in a few clicks and questions. The only requirements are queries from the users. The Chat bot detects the intent of the queries and accordingly parses the website to find the related information.

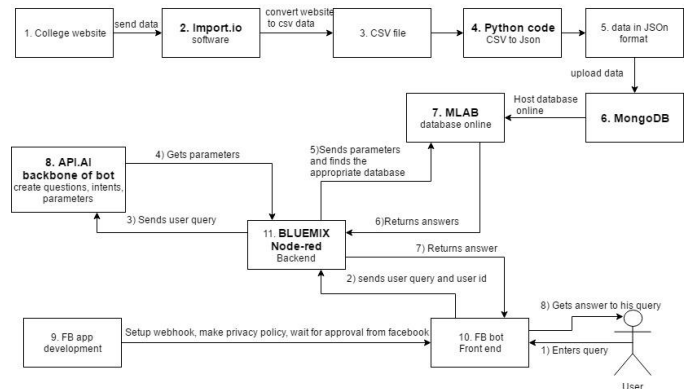


Figure 2. Proposed Architecture Diagram

**MODULES:**

**Import.io:**

Import.io is an information retrieval tool. It is the activity of obtaining information resources relevant to an information need from a collection of information resources. Automated

information retrieval systems are used to reduce what has been called "information overload". Many universities and public libraries use IR systems to provide access to books, journals and other documents.

API.ai:

API.ai is an API that makes it easy for developers to build bots, applications and devices that you can talk or text. It is a platform used to recognize the intent of the user using NLP and formulate a response for them using NLG.

Chatbot:

The users will give voice or text, you get back structured data. It's that simple. Chatbot will consist of Natural language processing, Natural Language generation and interaction with Wit.ai for intent.

Database:

The information retrieved from the college website is stored in a Database or knowledge base. This database is used for keyword matching. Each keyword uses a matching option to help control which searches should trigger your ad to show. You can choose one or more matching options for a keyword. If you don't specify a matching option, keywords are considered as broad match.

UI/UX:

This is graphical user interface which the user will use to interact with the chatbot. The college website will consist of a chat module which will serve as the UI for the users. The Chat module will accept the queries from the users and revert the replies from the chatbot in a language which users understand. These bots will ultimately be unleashed on a) all the visible digital data in existence, b) apply vast computing power and cutting-edge algorithms to make sense of it all, and c) provide the ability to use this knowledge to converse with us about the world in a deeply meaningful way.

#### IV. IMPLEMENTATION

**Setting up the intents and entities in API.AI:** API.AI is a natural language understanding platform that makes it easy for developers (and non-developers) to design and integrate intelligent and sophisticated conversational user interfaces into mobile apps, web applications, devices, and bots. Our goal is to make the process of creating and integrating sophisticated conversational user interfaces as simple as possible. Entities represent concepts and serve as a powerful tool for extracting parameter values from natural language inputs. The entities that are used in an agent will depend on the parameter values that are expected to be returned because of agent functioning. In other words, a developer need not create entities for every concept mentioned in the agent – only for those required for actionable data. Intent represents a mapping between what a user says and what action should be

taken by your software. Intent interfaces have the following sections:

- User says
- Action
- Response

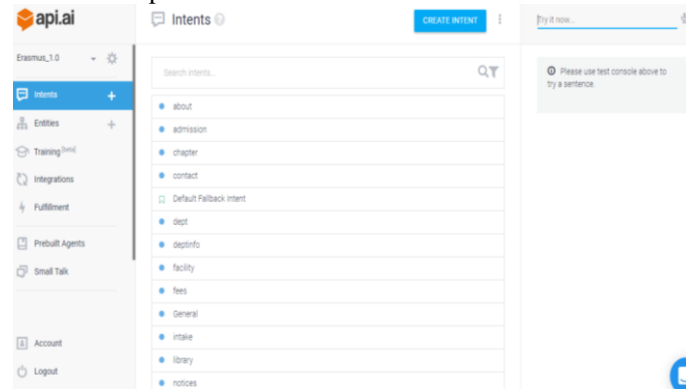


Figure 3. User Interface of Api.ai

**Extracting data from college website:** The college website consists of a lot of data in various formats, e.g.: text, links, pdfs etc. The users require this data as answers to their queries. The data is extracted from college website using Import.io. **Import.io** is a very powerful and easy-to-use tool for data extraction that has the aim of getting data from any website in a structured way. It is all about web scraping and extracting data from the web without using any code whatsoever. We scrape the web to extract this data and use it to make decisions. It scrapes data from website in CSV format.

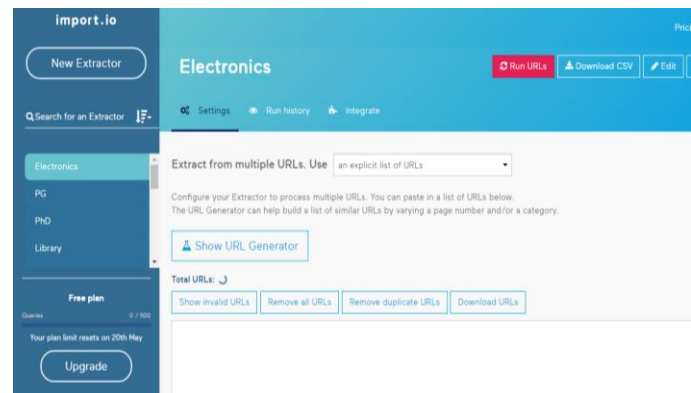


Figure 4. Screenshot of Import.io

**Converting the data to JSON format:** After extracting the data from website in CSV format, we need to convert them into JSON format. API.AI accepts inputs in JSON format and gives outputs in JSON format. Hence, data in JSON format is the best way to retrieve results based on user query.

**Upload the data on an online database:** The data in JSON Format must be uploaded in an online database system, which can be accessed to retrieve results based on parameter and intents. We use MongoDB in this case. MongoDB is a free

and open-source cross-platform document-oriented database program. It is classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. *mLab's* MongoDB hosting platform is the fastest growing cloud Database-as-a-Service in the world. Get started with a free database and expert support. mLab is a fully managed cloud database service that hosts MongoDB databases. mLab runs on cloud providers Amazon, Google, and Microsoft Azure, and has partnered with platform-as-a-service providers. We upload the JSON files as collections under a mlab database.

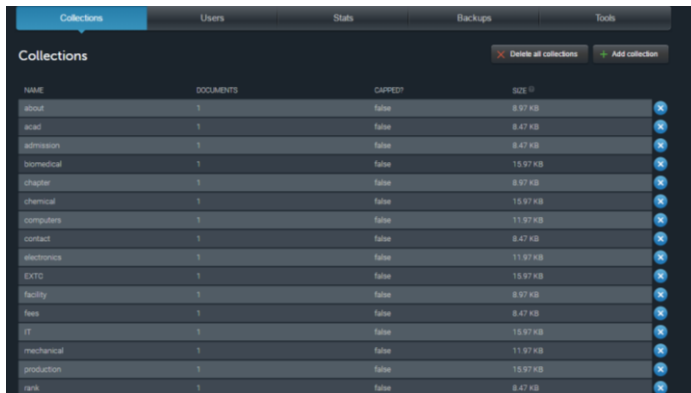


Figure 5. MongoDB interface

**Setting up a Facebook Messenger app:** A Facebook Chat Messenger application must be developed as the user interface or the chat module where user can ask their queries. This can be done on developers.facebook.com, which the Facebook developer platform to build and deploy apps on Facebook. After developing the app, it must be sent to Facebook review authorities which test the app thereafter giving permissions to deploy the app for Facebook users.

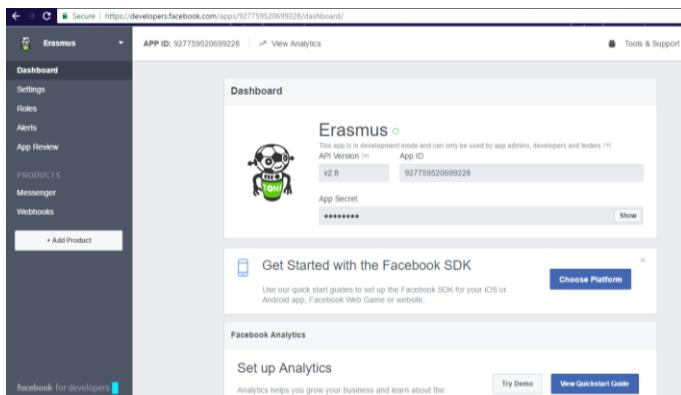


Figure 6. Facebook Developers Page

**Setting up Webhook for API.AI and Facebook Messenger:** Webhook integration allows you to pass information from a matched intent into a web service and get a result from it. A Webhook is an HTTP callback: an HTTP POST that occurs when something happens; a simple event-notification via

HTTP POST. A web application implementing Webhooks will POST a message to a URL when certain things happen. IBM Bluemix is a platform-as-a-service environment developed by IBM. It supports several programming languages and services as well as the DevOps development methodology in an integrated way to create, execute, deploy and manage applications in the cloud. We use IBM Bluemix's Node-red services to develop set up a webhook which uses facebook's access token to connect to api.ai's agent. It accepts queries from Facebook users. It converts these queries into messages which it sends to api.ai agent. The agent parses the queries and picks up the intents from the query. It does this using In-built Natural Language Processing algorithms. The agent then responds with parameters based on user intents and entities which are set.

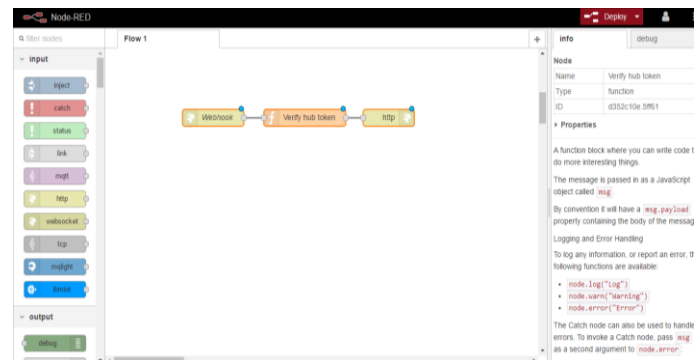


Figure 7. Backend Development on IBM Bluemix

**Using parameters acquired from API.AI to retrieve answers from online database:** IBM Bluemix is a platform-as-a-service environment developed by IBM. It supports several programming languages and services as well as the DevOps development methodology in an integrated way to create, execute, deploy and manage applications in the cloud. We use IBM Bluemix's Node-red services to develop set up a webhook which sends the parameters obtained from api.ai's agent. These parameters can be used to select the appropriate collection from online MongoDB database. The required answers are retrieved using the parameters and sent to API.AI.

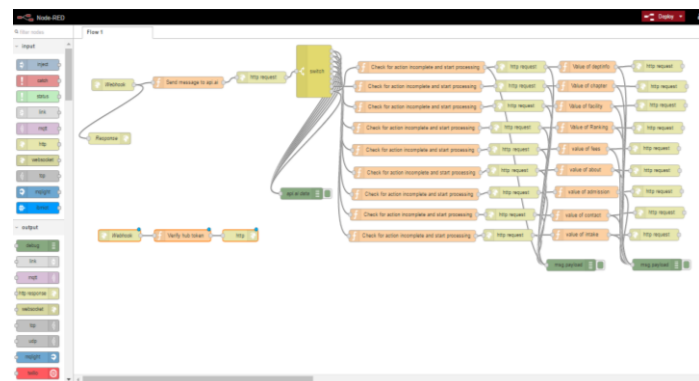


Figure 7. Backend Development on IBM Bluemix

**Sending the answer back to the user:** Natural Language Generation is applied on the answers retrieved from the online databases. The answer generated is then sent back to the user using the same webhook between api.ai and Facebook.

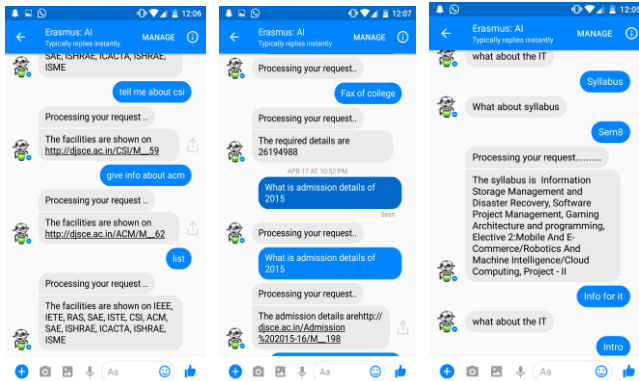


Figure 8. Working Chatbot Demo

#### IV. CONCLUSION

Through the proposed project and research, we draw a conclusion that the new proposed system will help the college system by overcoming some major flaws in the already present chatbot systems. Using the above study, we can further enhance the chat bot to help the college facility cater to anyone who visits the college website. This project will alleviate the pitfalls that the staff of the college comes across during the peak times of admissions

#### V. REFERENCES

- [1] Sameera A. Abdul-Kader, Dr. John Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 7, 2015
- [2] Bayan Abu Shavar, Eric Atwell, "Chatbots: are they really useful?", LDV-Forum 2007, Band 22(1) – 29
- [3] B. P. Kiptonui, "Chatbot Technology: A Possible Means of Unlocking Student Potential to Learn How to Learn, Educational Research", Vol.4, Issue.2, pp.218-221, 2013.
- [4] M. Dahiya, "A Tool of Conversation: Chatbot", IJSCE, Volume 5, E-ISSN: 2347-2693, pp.1-2, May 2017.
- [5] J. Jia, "The Study of the Application of a Keywords-based Chatbot System on the Teaching of Foreign Languages", Report of University of Augsburg, Augsburg, pp.1-36, 2003.
- [6] Bonnie Chantaratwong, "The Learning Chatbot Bonnie", IMS-256 Final Project, Fall 2006
- [7] Johan Rahman, "Implementation of ALICE chatbot as domain specific knowledge bot for BRACU(FAQbot)"
- [8] Shah, Rishabh, Siddhant Lahoti, and K. Lavanya. "An intelligent chat-bot using natural language processing." International Journal of Engineering Research 6.5 : 281-286. 2017.

#### Authors Profile

*Mr. Jash Thakkar* pursued his Bachelor of Engineering in Information Technology from Dwarkadas J. Sanghvi College of Engineering in 2017. He is currently working at ZS Associates, Pune as a Business Technology Analyst. He specializes in Business and Data Analytics, Business Consulting.

*Prof. Purva Raut* pursued her M.Tech in Computer engineering from NMIMS University, Mumbai. She has been working as an assistant professor at Dept of Information Technology, Dwarkadas J. Sanghvi College of Engineering for 9 years. Her main research work focuses on Intelligent Systems and Distributed Systems.

*Mr. Yash Doshi* pursued his Bachelor of Engineering in Information Technology from Dwarkadas J. Sanghvi College of Engineering in 2017. He is currently working at PriceWaterhouseCoopers, Mumbai as a Senior Analyst. He specializes in Penetration testing, IT Auditing and Compliance Review.

*Ms. Krishma Parekh* pursued her Bachelor of Engineering in Information Technology from Dwarkadas J. Sanghvi College of Engineering in 2017. She worked at Reliance, Mumbai as a BI Analyst for 10 months. She is currently pursuing her Masters in Management Information Systems from Carnegie Mellon University, Pennsylvania.