A SMART CHAT-BOT

Final Report for the MCA Project

Submitted by
AHMAD ALI

(15SCSE111003 / 1513111002)

In partial fulfillment for the award of the degree of

Integrated BCA+MCA

Computer Application

SCHOOL OF COMPUTING SCIENCE & ENGINEERING

Under the Supervision of

Mr. M.ARVINDHAN

Asst. Professor

School of Computing Science & Engineering

April / May - 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report “A SMART CHAT-BOT” is the bonafide work of “AHMAD ALI (1513111002) ” who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr. MUNISH SHABARWAL PhD (Management),PhD (CS)	**M.Arvindhan**
Professor & Dean,	Asst. Professor
School of Computing Science & Engineering	School of Computing Science & Engineering

ACKNOWLEDGMENT

I wish to record my deep sense of gratitude and profound thanks to my research supervisor **M.Arvindhan**, Asst. Professor, School of Computing Science and Engineering department, Galgotias University, Greater Noida for his keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this dissertation into fruition.

I extend my sincere thanks to Dean SCSE for providing excellent platform and resources to carry out my research projects. In addition, I would like to thank the panel members for their valuable suggestions and support during presentation of my research projects.

Finally, I extend my sincere thanks to the University Management, All faculty members, non-teaching staff members and Lab Assistants of the SCSE Department, Galgotias University, Greater Noida, for their valuable support throughout the course of my M.C.A Dissertation.

I thank my friends, fellow researchers and family members who have encouraged me in my research efforts and shouldered me in needy times.

AHMAD ALI

1513111002

15SCSE111003

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the MCA Dissertation Final Report, entitled '**A Smart Chat-Bot**' in Computer Science & Engineering and submitted in the School of Computing Science Engineering of the Galgotias University, Greater Noida is an authentic record of my own work carried out during a period from JAN 2020 to MAY 2020 under the supervision of **M.Arvindhan**, professor of School of Computing Science & Engineering, Galgotias University, Greater Noida. The Content presented in the dissertation has not been submitted by me for the award of any other degree of this or any other Institute.

AHMAD ALI
1513111002

TABLE OF CONTENTS

CHAPTER NO. **PAGE NO**

ABSTRACT

There are varieties of User interfaces for software applications. For eg. Command-line, Graphical User Interface, Web Application, and even Voice. The most popular user interfaces are GUI and Web-based applications, Sometimes we do need an alternative interface. For example, For Concurrent connectivity, or Instant attending of clients query, a chat bot based interface may suit the need. The Chat-Bots provide a text-based user interface, Which allows the user to type commands or queries and receive responses in form of text as well as text to speech.

Chat bots are usually a stateful services, remembering previous commands (and perhaps even conversation) in order to provide functionality. When chat bot technology is integrated with popular web services it can be utilized securely by an even larger audience. Chatbots receive increasing attention from media and industry, but at the same time it is not yet well known what Chat-Bots really are, what they can be used for and how to create them.

The goal of this work is to answer these three questions by analyzing existing platforms, products and technologies, and additionally developing an exemplary chatbot. Explaining what chatbots are, demystifying what to use them for and showing how to create them will help more people to be able to use and create chatbots and thereby accelerate the development of the chatbot ecosystem. This chatbot project is built using artificial algorithms that analyzes user's queries and understand user's message. This System is a web application which provides answer to the query of the person.

We just have to query through the bot which is used for chatting. We can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query. The answers are appropriate , according to what the user queries are. The System analyzes the question and then answers to the user. The system answers to the query as if it is

answered by the person. With the help of artificial intelligence, the system answers the query. We will be using Python v2 for the back-end.

2. INTRODUCTION

This document will provide all of the in depth details for the project, A Smart Chat-Bot. It will serve as a reference for developers during the development of the final version of the system.

(i) Overall Description

A Smart Chat-Bot is an automated Chat-Bot that receives queries from its users and tries to understand the query, and provide answers accordingly Chat bots will remember previous commands and conversation, in order to provide functionality. When chat bot technology is integrated with popular web services it can be utilized securely by an even larger audience. Chat-Bots receive increasing attention from media and industry, but at the same time it is not yet well known what Chat-Bots really are, what they can be used for and how to create them.

(ii) Purpose

- Anytime Accessibility : On an average people spend around 7 minutes until they are assigned to a person. So to keep them occupied in earlier minutes, It'll do great job.
- Handling Capacity : Unlike humans who can only communicate with one human at a time, chat bots can simultaneously have conversations with thousands of people. No matter what time of the day it is or how many people are contacting you, every single one of them will be answered immediately.
- Cost Effective : Hiring a human for a job is never a cheap affair, So using Chat-Bot for specific tasks, The Organization or Business can cut the Cost.
- Work Automation : People tend to be less productive when given a recurring job or work. We humans usually get bored doing the same thing over and over again. ChatBots can now automate tasks which are to be done frequently and at the right time. This helps people save time and be more productive.
- Personal Assistant : People can use Bots as personal advisor for clothing recommendations, or ask trading tips from a finance bot, suggest places to visit from a travel bot and so forth. This would help the users get a more personal touch from the chatbot. Also, the chatbot will remember all your choices and provide you with relevant choices the next time you visit it.

(iii) Project Scope

A Smart Chatbot is an automated chatbot that receives queries from its users and tries to understand the query, and provide answers accordingly. It is done by converting the given sentence(which is in English) into a query easily understood by machine, after which it goes through the relevant datasets (stored in xml form) to find the information relevant to it or generates one of itself, and finally it returns the answer in the form of sentence (also in English). In simple words, it can answer the given questions like we do, instead of finding a bunch of websites that contains the answer (like searching Google). For better understanding, when we input a sentence let say "Hi", it will greet with a response “Hello there ”. Our main objective is to create a Web API, and a simple web interface that demonstrates the use of the API. Also, it will be able to provide an interactive way , which is also simpler to have queries answered, and to have a virtual assisstant as well .

3. Literature Survey

3.1 Product Perspective

A Smart Chatbot, however, will try to understand the query and provide a definitive answer. There will be four main units to the system working together to understand the query and return an appropriate answer:

3.1.1 Generic question construction – It will take natural language question and making it more generic.
3.1.2 Generic answer construction - It will take a generic question template and provide a generic answer template.
3.1.3 Generic answer population – It is capable of taking a generic answer template and populating it with information from the dataset to form an answer[2].
3.1.4 Information extraction – It is capable of finding information through structured datasets stored using aiml in xml form.[3][4]

3.2 Product Features

The major features for A Smart Chatbot will be the following:

3.2.1 Web API – An API call will include a question in the form of a query string url parameter and the service will reply.
3.2.2 Natural Language Processing - The system will take in questions written in standard English.[6]
3.2.3 Natural Language Responses - The answer to the question will be written in standard and understandable English.[6]
3.2.4 Information Extraction: There will be datasets containing all the information needed, populated

3.3 User Classes and Characteristics

The two classes of users for this system are described below:

3.3.1 API users

API users consist of application developers who want to incorporate A Smart Chatbot API into other software applications.

3.3.2 Web app users

These users consist of non-technical users who want to get answers for their questions. These users ask questions and can get answers with mobile, web, or text messaging interfaces. This class of users include A Smart Chatbot's current prospective students, teaching faculty, and staff.

4. PROBLEM STATEMENT

4.1 Limited Question Scope

Creating a chatbot able to answer every single question is not possible to implement with current technology and within the duration of the project, so the system will be able to answer questions about limited topics.

4.2 Language

The system will only support questions in standard English.

4.3 Datasets

Giving support for wide varieties of Topics will require writing more and more modules, Which is quite challenging for doing Individually.

5.PROPOSED MODEL

5.1 Architectural Design

5.1.1 Subsystems

Front End Application: Web app which receives question from the user and talks to the Smart Chatbot API Service to give the answer.
Smart Chatbot API Service: It receives HTTP GET requests containing user queries and forwards them to Smart Chatbot Main. [7]
Smart Chatbot Main: Main process, which coordinates the other subsystems.
Generic Question Construction: It takes the question from the user, and creates a generic question template by replacing certain nouns with generic representations.
Generic Answer Construction: It takes in a generic question template and outputs a generic answer template.
Generic Answer Population: It takes a generic answer template and populates it with information from the dataset to form an answer.
Error Handler: If an error occurs during execution of the system, an exception will be thrown, and the error handler will decide how to respond to the user.
Dataset(aiml)[3]: It stores information needed by question construction and generic answer[1].

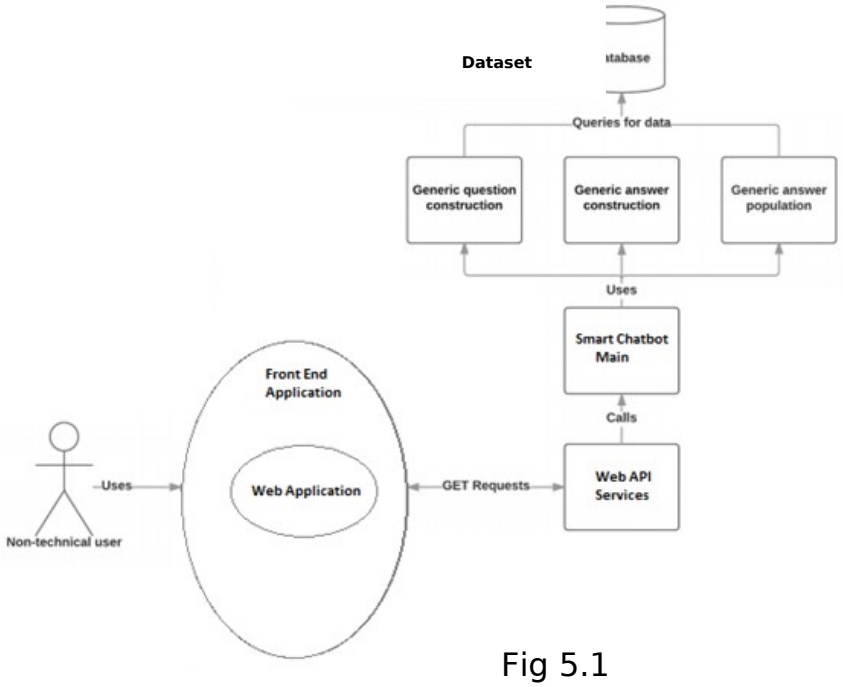


Fig 5.1

5.1.2 System operation

Following figure illustrates the interactions of the system's subsystems during the typical use of a front end application.

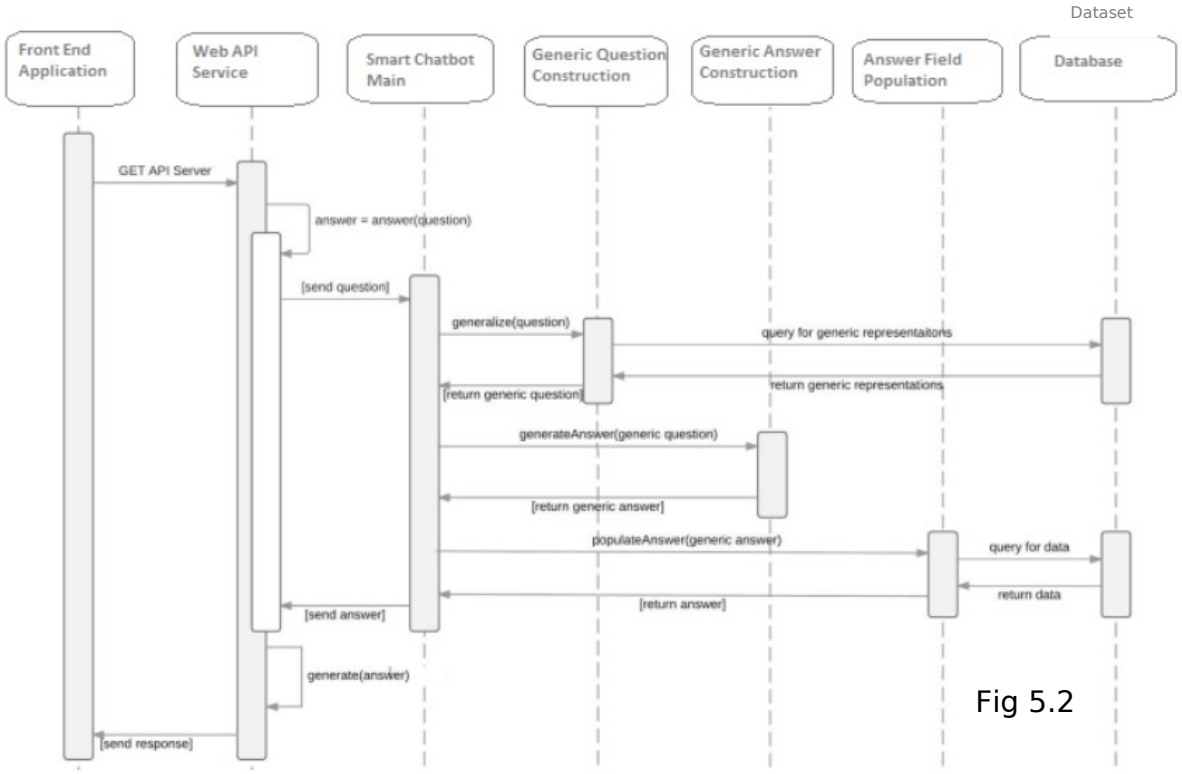
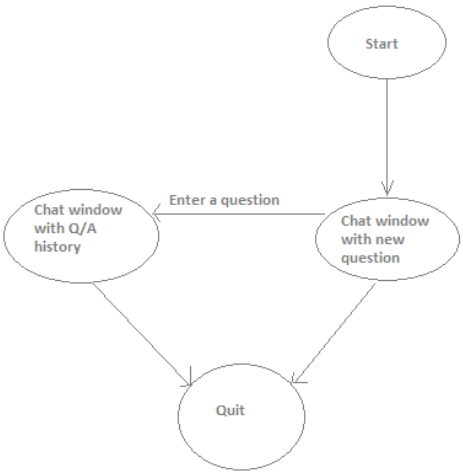


Fig 5.2

5.2 Activity Diagram



5.3 Main Description

5.3.1 Smart Chatbot Main

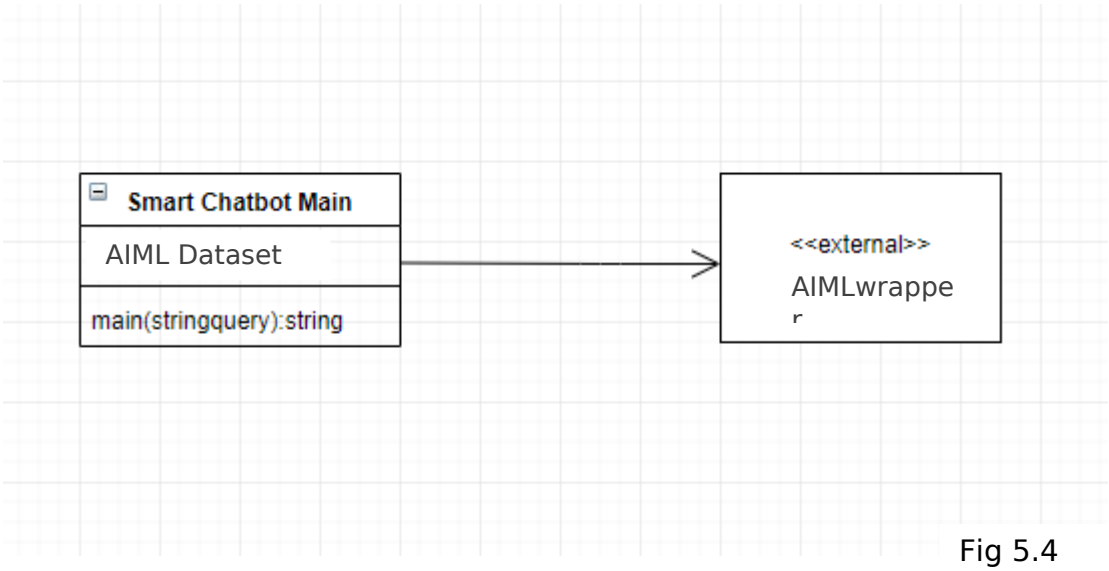


Fig 5.4

-

5.3.2 Error Handler

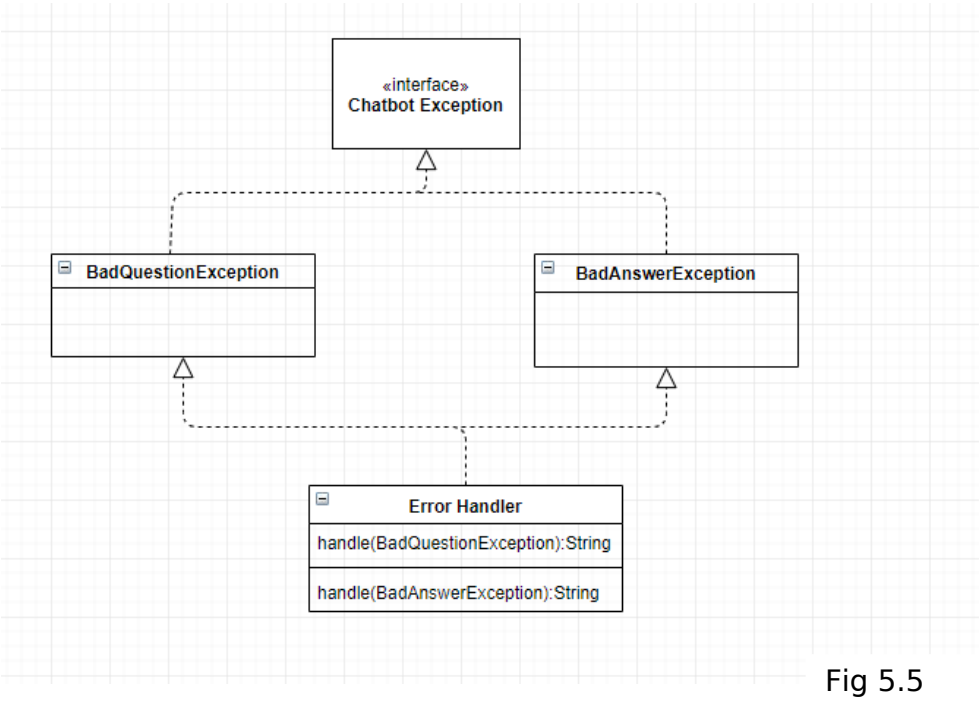


Fig 5.5

5.3.3 Generic Answer Construction

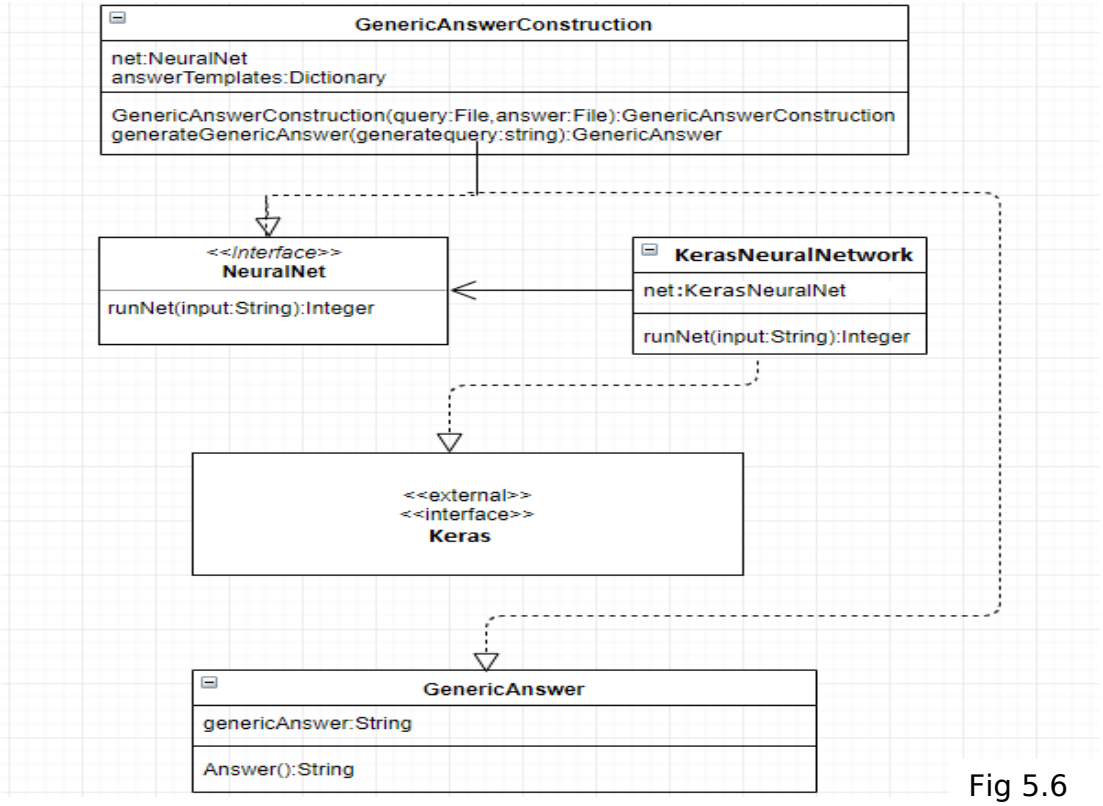


Fig 5.6

5.4 Design Rationale

In this section, we explain our choices for the following designs and systems with our reasons behind them.

5.4.1 Generic Question Answering System

We opted for a generic question answering system instead of specific questions and answers to eliminate the amount training neural network has to perform. In addition, if some answers change, we would not have to train the neural network all over again.

5.4.2 Python

We selected Python as our main programming language because it has the highest number of libraries in the Artificial Intelligence field, and in future it will be used mostly for AI.

5.4.3 API

Since Smart Chatbot focuses on researching on question answering system, we decided to create a web API that demonstrates our findings and benefits to other application developers. Sample web applications will be developed to showcase our result, but we are not focused on developing any comprehensive, robust front-end systems.

5.4.4 Brain

There are two ways to create a chatbot. First, we write all the logic that would be able take any question and give an appropriate response. This would be adding a lot of rules manually and it will take a lot of time and effort. It will also not be reusable for any other domain. The second method is machine learning. Using neural network would allow us train the neural net with a set of question and answers and if a new question is passed to the neural net, the neural net will try to provide an appropriate answer.[3]

5.4.5 Modularize

Since we are not sure how well some algorithms will perform, modularizing subsystems will allow us to retire poorly-functioning algorithms and introduce new ones without breaking the entire system.

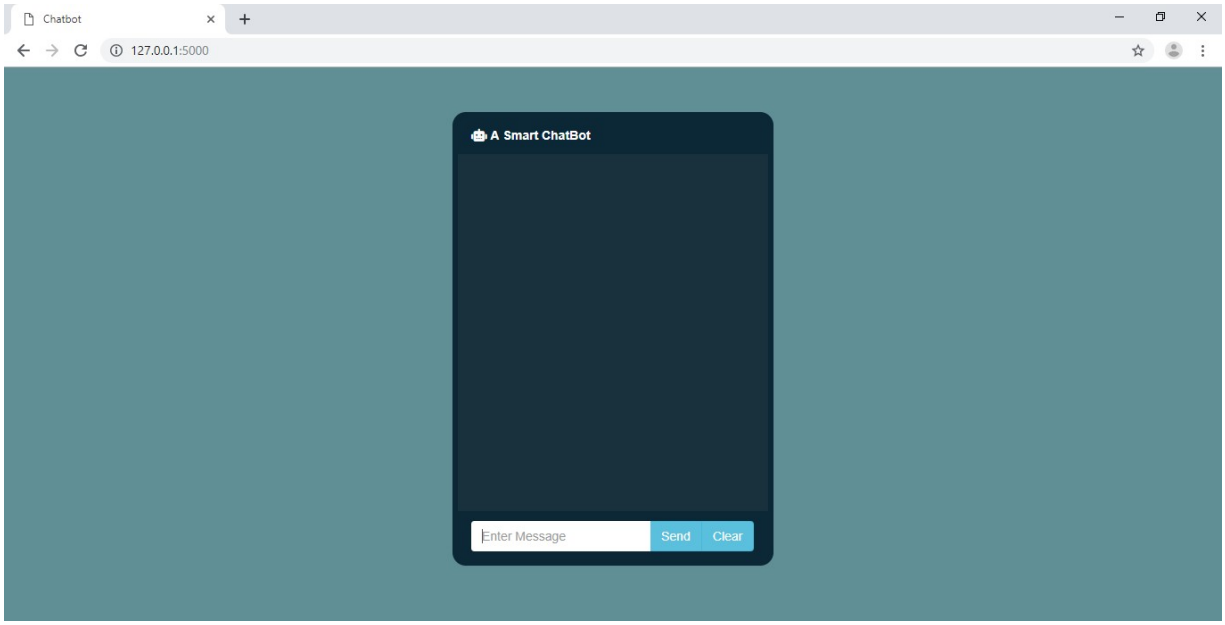
5.4.6 Error Handler

We added error handler to process ambiguous questions entered by the user and unexpected responses returned by the neural network. In addition, the error handler lets us provide support for additional error cases efficiently, by just creating a new exception type and handle function.

5.5 Data Description

Smart Chatbot uses Aiml datasets stored in the form of xml needed by generic answer construction and generic answer population. The datasets essentially provides the data needed for all questions that Smart Chatbot can handle.Our data is also going to be trained .[5]

5.6 User Interface Design



6. IMPLEMENTATION

Functional Requirements

6.1 API Calls

6.1.1 Client Responsibilities

6.1.1.1 The client will send a GET request to the Web API with the question as a URL parameter.

6.1.1.2 The client will specify the header Content-Type: application in their requests by default.

6.1.1.3 A valid API query is a single URL parameter containing one sentence that is a question in standard English.

6.1.1.4 The server will reply with either data or an error.

6.1.2 Server Responsibilities

6.1.2.1 The server will send all API data in JSON response documents.

6.1.2.2 The server will respond with a Bad Request status code if a request does not specify.

6.1 Generic Question Construction

6.2.1 Input & Output Format

6.2.1.1 This unit will receive a text string from the URL parameter.

6.2.1.2 It will identify important words in the sentence and replace them with generic representations preceded by an escape character

6.2.1.3 It will output the sentence as a string

6.2.1.4 It will output a map of generic representations to the words they replaced and also.

6.2.2 Error Handling

An error during this process means there was a problem parsing the sentence and creating the generic question. In this case, return a message such as “Sorry, I didn’t understand that.”

6.1 Generic Answer Construction

6.3.1 Input & Output Format

6.3.1.1 This unit will receive the output sentence from the Generic Question Construction unit as input.

6.3.1.2 It will generate a generic answer sentence using the input.

6.3.1.3 It will output the generic answer sentence.

6.3.2 Error Handling
If there was an error here, then the unit failed to create a generic answer given a generic sentence. In this case, simply fallback to the error handling.

6.1 Generic Answer Population

6.4.1 Input & Output Format

6.4.1.1 This unit will receive as input a mapping from the Generic Question Construction.
6.4.1.2 It will receive the string as input a generic answer from the Generic Answer Construction.
6.4.1.3 It will query the datasets for data about the elements in the mapping.
6.4.1.4 It will replace the representations in the generic answer with data.
6.4.1.5 It will output the answer to the original question.

6.4.2 Error Handling
If querying the AIML dataset did not provide an answer, the system will say that it does not have an answer.

6.1 User Interfaces

6.5.1 Website application GUI

6.5.1.1 The GUI will have a textbox that will accept inputs from a keyboard
6.5.1.2 It will have a “Send” button which sends text from the textbox to the API when clicked.
6.5.1.3 It will have a chat window displaying questions sent to the system and responses from the API.
6.5.1.4 The chat window will contain all questions and answers from the current session, with a scroll bar if all messages can’t fit on the screen.
6.5.1.5 If there is a network issue, the chat window will display an error message.

7. IMPLEMENTATION

NON-FUNCTIONAL REQUIREMENT

7.1 Modularity

The system will be designed in such a way that the algorithms for the four main units will be able to be easily swapped out.

7.2 Accuracy

7.2.1 The overall accuracy of the Web API's response will be measured using a developer-made testing set.
7.2.2 The overall accuracy is calculated by dividing total number of correct answers by the number of questions asked.
7.2.3 The accuracy of the Generic Question Construction part will be close to 50%-80%.
7.2.4 The accuracy of the Generic Answer Construction unit will be close to 50%-80%.
7.2.5 The accuracy of the Generic Answer Population unit will be close to 50%-80%.

Glossary

Chatbot: An interface, usually text based, specializing in the mimicry of natural language conversation aka “artificial conversational entity.”
Python : Python is an interpreted high-level programming language for general-purpose programming.[8]
GUI: Graphic User Interface, a type of user interface that allows users to interact with the software through graphical icons (e.g. buttons, etc.).
HTML: Hypertext Markup Language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on webpages.
JSON: JavaScript Object Notation, a data-interchange format that is commonly used in exchanging data over the Internet.
Standard English: the language that can be understood by English-speaking high school graduates.
URL: Uniform Resource Locator, an address to a resource on the Internet.
URL parameter: parameters whose values are set in a webpage’s URL.
Web API: an application programming interface (API) for a web server.[7]
Aiml[1]: AIML is an XML based markup language meant to create artificial intelligent applications.
CSS: Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.

REFERENCES

1. AIML (2002). A.L.I.C.E AI Foundation. https://pandorabots.com/docs/
2. Bayan Abu Shawar (2003) https://www.researchgate.net/publication/267575064_Using_dialogue_corpora_to_train_a_chatbot
3. Chatbot (2015). https://www.chatbots.org/
4. R.Wallace (2003). The Elements of AIML Style, ALICE A.I Foundation.
5. Maria das Graças Bruno Marietto (2013). ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE: A BRIEF TUTORIAL
6. Metaguide (2011). https://meta-guide.com/software-meta-guide/100-best-ai-nlp-resources-aiml
7. Flask (2009). http://flask.pocoo.org/
8. Python (2006). https://www.tutorialspoint.com/python/
9. Html (2006). https://www.tutorialspoint.com/html/