AI CHATBOT FRAMEWORK AS A SERVICE

A PROJECT REPORT

Submitted by

MOHAMMED RASIK

REG NO: TKM20MCA2023

to

APJ Abdul Kalam Technological University

In partial fulfilment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report **AI CHATBOT FRAMEWORK AS A SERVICE**, submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Dr. NADERA BEEVI S**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Mrasik

Kollam

12/07/2022 MOHAMMED RASIK

DEPARTMENT OF COMPUTER APPLICATIONS
TKM COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that, the report entitled **AI CHATBOT FRAMEWORK AS A SERVICE**, submitted by **MOHAMMED RASIK**, to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor Head of the Department External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of our project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, for providing us with best facilities.

I would like to thank my project guide **Dr.Nadera Beevi S**, Department of Computer Applications, who motivated me throughout this project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout our course of study.

I owe my thanks to our friends and all others who have directly or indirectly helped us in the successful completion of this project.

MOHAMMED RASIK

ABSTRACT

This project aims at developing an AI Chatbot Framework which can be used to create chatbot applications for any type of business concern or entity. The application by default will not contain any intents or stories, however new intents and chat sequences can be created and trained by a non-technical user without any programming effort.The application provides a UI for creating, training and managing chat intents.

The Framework provides an easy interface for 3rd party API integrations, thereby enabling the chatbot to access any service or API for enhancing the user stories. The Framework provides interface for various HTTP methods such as GET, POST, PUT, DELETE and so on. The API response and failure is also managed efficiently in the framework.

The Chatbot Framework itself is an API service which can be consumed by another application as a service through a simple http call. Messaging services such as Facebook, Messenger, WhatsApp, Gmail and ordinary chat boxes and so on can easily plug in this API and enable the chatbot.

The application will be built using python Flask & Angular for back-end & front-end respectively. It will also use Spacy NLP & NLTK libraries along with TensorFlow modules for seamless Natural Language Processing. The stack will also use mongo DB for persistence.

Contents

List of Figures

Chapter 1

INTRODUCTION

The aim of this project is to develop an AI Chatbot Framework which serves as a platform for creating chatbots for any given business use cases.

A common end user without any programming knowledge can easily create intents, train the bot and create his own chatbot according to his/her needs.

Ex. This raw platform can be used by a hotel owner to add intents/use cases for hotel business such as menu, order food etc.

1.1 Problem Definition

Even though there are many chatbots and chat services available, most of them are either very sophisticated to use, or even if user friendly very costly such that small businesses cannot afford them or may not have the technical expertise to utilise them.

Most of the well-designed chatbots are proprietary, hence there is a need for developing more open-source solutions.

1.2 Objective

Create a Chatbot platform which is user friendly with UI and tools which can help the user to create his/her own intents and develop their own chatbot.

The chatbot should be available as an API service for other chat applications.

The chatbot should have capability to connect to 3rd party services.

The bot should be built with open-source tools such that there are no proprietary dependencies.

Chapter 2

LITERATURE SURVEY

A literature review is a complete examination and analysis of literature on a certain topic. When research questions are identified through a literature review, one seeks to answer them by looking for and analysing relevant material. Re-analysing the study’s results can lead to fresh discoveries, which is why literature reviews are important. A literature review summarises and explains the whole and current state of knowledge on a topic as found in academic books and journal articles. At university, there are two types of literature reviews: one that students are asked to write as a standalone assignment in a course, and another that is prepared as an introduction to, or preparation for, a larger work, usually a thesis or research report. The type of review you are writing will affect the focus and perspective of your review, as well as the type of hypothesis or thesis argument you make. Reading published literature reviews or the introductory chapters of theses and dissertations in your own subject area is one approach to comprehend the differences between these two forms.

2.1 Related Works

Chatbot (and voicebot) applications are increasingly adopted in various domains such as e-commerce or customer services as a direct communication channel between companies and end-users. Multiple frameworks have been developed to ease their definition and deployment. While these frameworks are efficient to design simple chatbot applications, they still require advanced technical knowledge to define complex interactions and are difficult to evolve along with the company needs (e.g. it is typically impossible to change the NL engine provider). In addition, the deployment of a chatbot application usually requires a deep understanding of the targeted platforms, especially back-end connections, increasing the development and maintenance costs. Xatkit framework is introduced and explained by Daniel et al. [1]. Xatkit tackles these issues by providing a set of Domain Specific Languages to define chatbots (and voicebots and bots in general) in a platform-independent way. Xatkit also comes with a runtime engine that automatically deploys the chatbot application and manages the defined conversation logic over the platforms of choice. Xatkit's modular architecture facilitates the separate evolution of any of its components. Xatkit is open source and fully available online.

Managing and evolving a chatbot's content is a laborious process and there is still a lack of standardization. In this context of standardization, the absence of a management process can lead to bad user experiences with a chatbot. Another work proposed the Chatbot Management Process, a methodology for content management on chatbot systems [2]. The proposed methodology is based on the experiences acquired with the development of Evatalk, the chatbot for the Brazilian Virtual School of Government. The focus of this methodology is to evolve the chatbot content through the analysis of user interactions, allowing a cyclic and human-supervised process. The proposed methodology is divided the proposed methodology into three distinct phases, namely, manage, build, and analyze. Moreover, the proposed methodology presents a clear definition of the roles of the chatbot team. It is validated the proposed methodology along with the creation of the Evatalk chatbot, whose amount of interactions was of 22,771 for the 1,698,957 enrolled attendees in the Brazillian Virtual School of Government in 2020. The application of the methodology on Evatalk's chatbot brought positive results: reduced the chatbot's human hand-off rate is reduced from 44.43% to 30.16%, the chatbot's knowledge base examples increased by 160% whilst maintaining a high percentage of confidence in its responses and keeping the user satisfaction collected in conversations stable.

Chatbot communication, in which a robot communicates with a human being in natural language in an open domain, has achieved significant progress. However, it still suffers from problems such as a lack of diversity and contextual relevance [3]. In this paper, A retrieval-polished (RP) model is proposed in the paper for response generation that polishes a draft response based on a retrieved prototype. In particular, we first adopt a prototype selector is first adopted to retrieve a contextually similar prototype. Then, a generation-based polisher is designed to obtain a polished response. Finally, we introduce a polished response filter to choose whether the final reply should be the retrieved response or the polished response. Extensive experiments on a dialog corpus show that our method outperforms retrieval-based and generation-based chatbots with respect to fluency, contextual relevance, and response diversity. Specifically, our model achieves substantial improvement compared with several strong baselines.

Chapter 3

METHODOLOGY

3.1 Proposed System

The proposed system consists of the AI Chatbot framework or the back-end API which facilitates the chatbot services. As shown in Figure 3.1, the end-user of customer uses a chat app to communicate with the chatbot. The chatbot receives the message from the user, it uses its trained data to identify the intent by breaking the keywords. Once the intent is identified, the chatbot collects necessary parameters and makes the API calls if required to 3rd party applications. The response collected is served to the end-user after processing it. The business user defines the intents inside the platform and trains the data, the data is stored in the Mongo database which is later retrieved for intent processing.

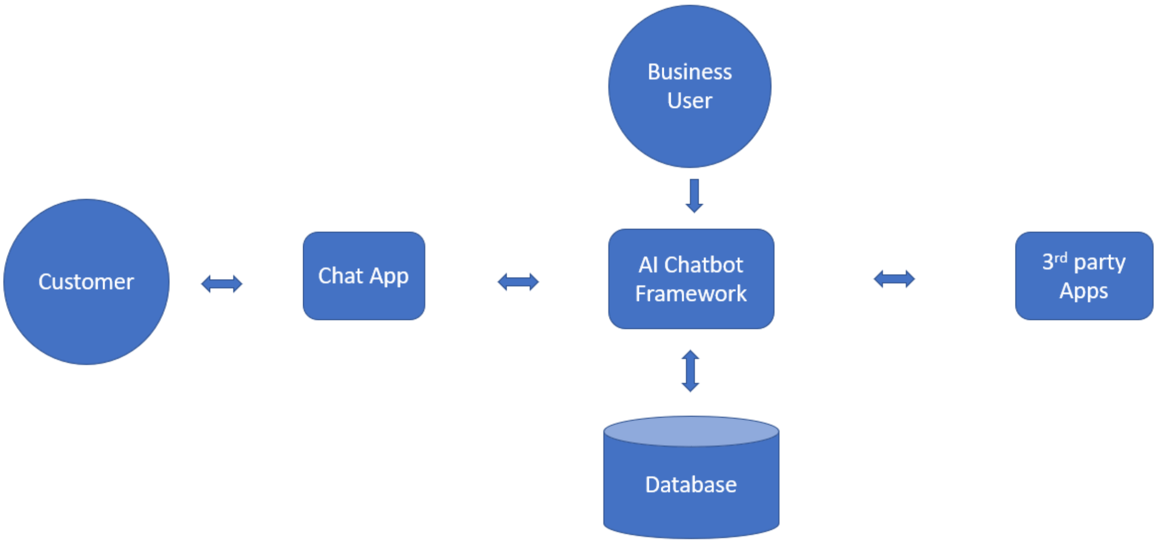
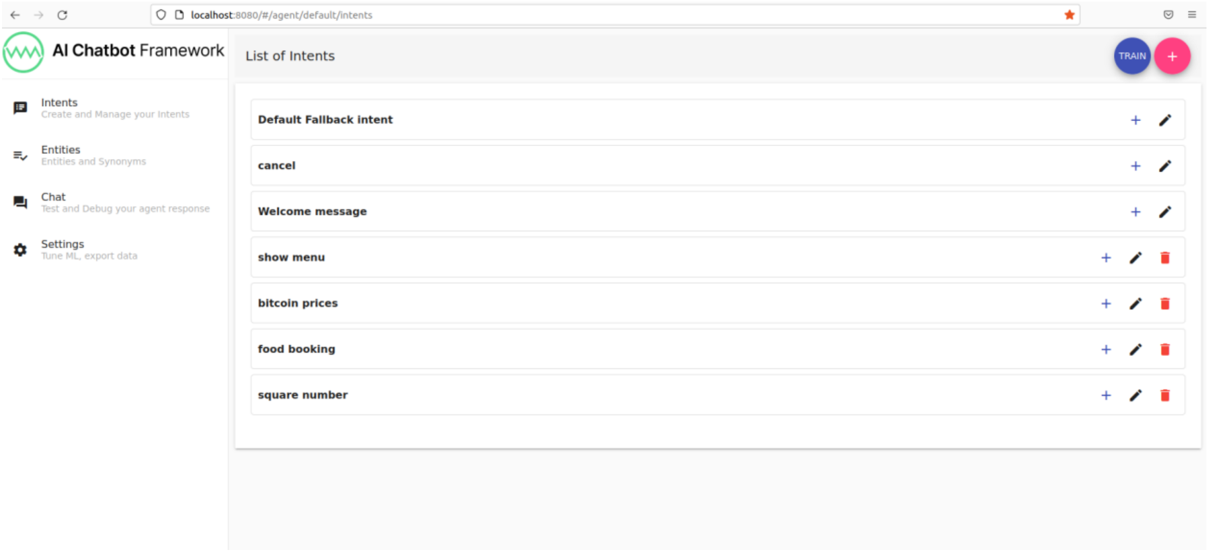


Figure 3.1: Block diagram of the proposed system

3.2 Graphical user interface

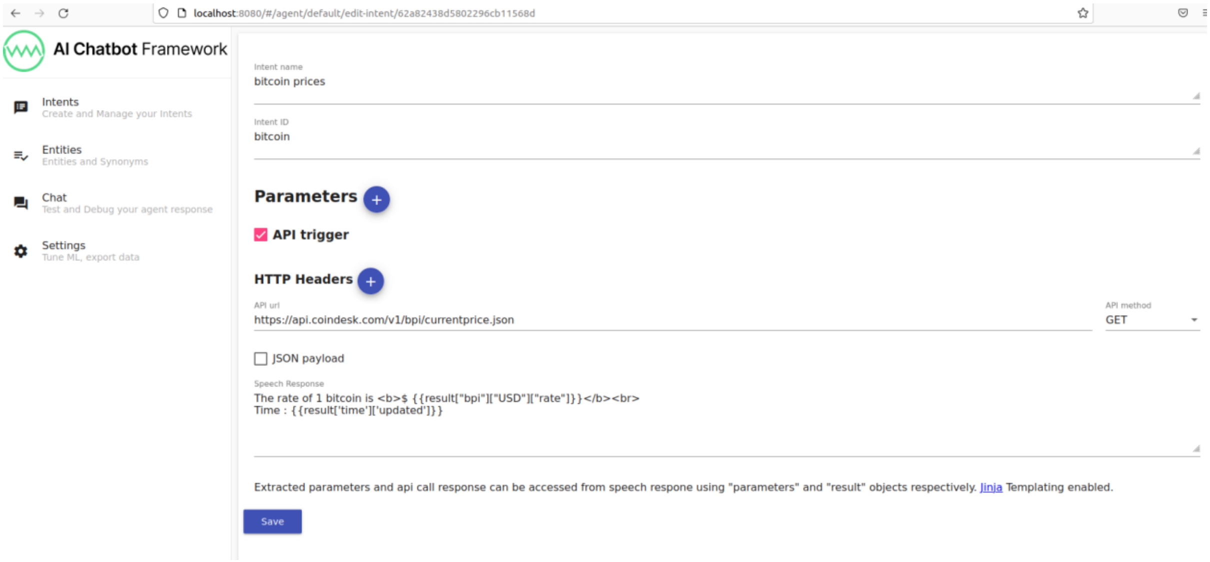
The Main Intents screen is where we can view all the intents in the chatbot, new intents can be created as well as existing ones can be edited as shown in Figure 3.2. Next the detailed intent creation screen as shown in Figure 3.3 where new intents can be created. The test chat area is shown in Figure 3.4 where trained intents can be accessed using chat. The training screen shown in Figure 3.5 helps in training the intent using example chat snippets. Finally Figure 3.6 shows the UI where platform settings such as confidence can be updated.


localhost:8080/#/agent/default/intents
AI Chatbot Framework
Intents
Create and Manage your Intents
Entities
Entities and Synonyms
Chat
Test and Debug your agent response
Settings
Tune ML, export data
TRAIN
+

List of Intents

- Default Fallback intent
- cancel
- Welcome message
- show menu
- bitcoin prices
- food booking
- square number

Figure 3.2: Main Intent Screen


localhost:8080/#/agent/default/edit-intent/62a82438d5802296cb11568d
AI Chatbot Framework
Intents
Create and Manage your Intents
Entities
Entities and Synonyms
Chat
Test and Debug your agent response
Settings
Tune ML, export data
Save

Intent name
bitcoin prices

Intent ID
bitcoin

Parameters +

☑ API trigger

HTTP Headers +

API url
https://api.coindesk.com/v1/bpi/currentprice.json

API method
GET

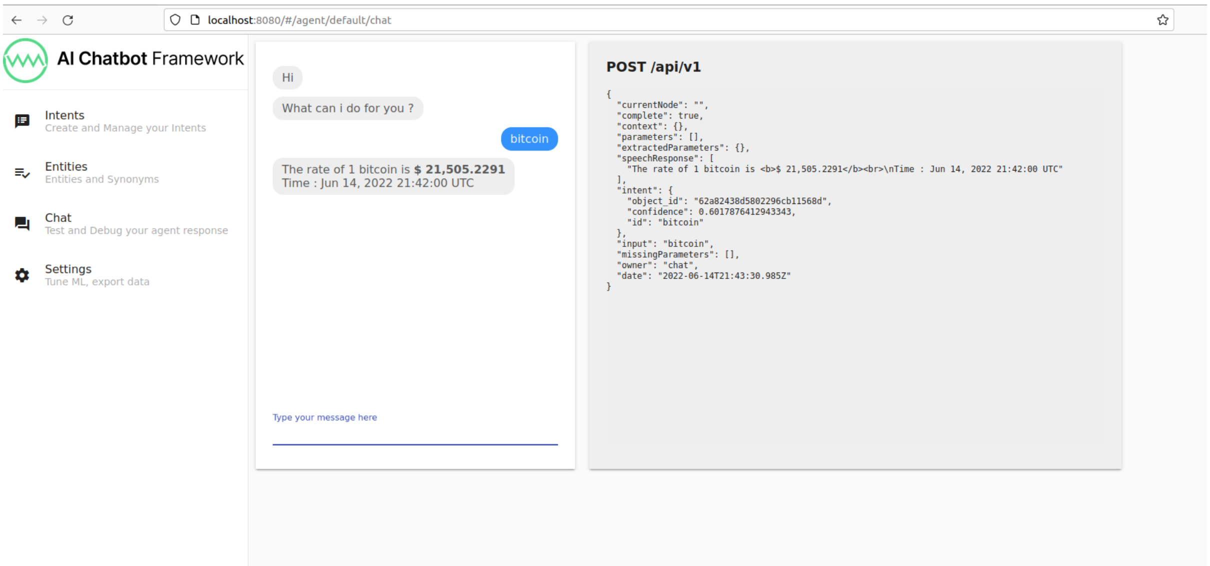
□ JSON payload

Speech Response

```
The rate of 1 bitcoin is <b>$ {{result["bpi"]["USD"]["rate"]}}</b><br>
Time : {{result['time']['updated']}}
```

Extracted parameters and api call response can be accessed from speech respone using "parameters" and "result" objects respectively. Jinja Templating enabled.

Figure 3.3: Intent Creation



Hi

What can i do for you ?

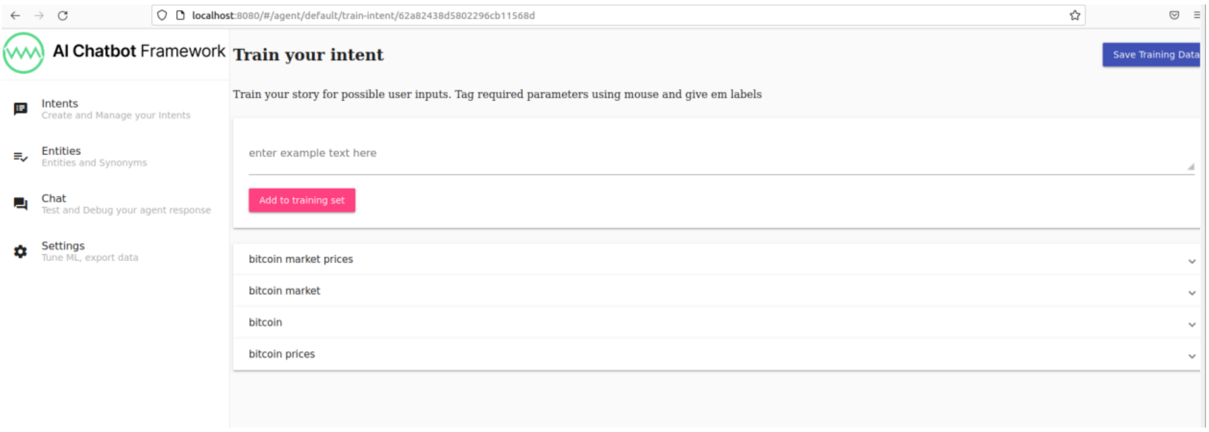
bitcoin

The rate of 1 bitcoin is **$ 21,505.2291**
Time : Jun 14, 2022 21:42:00 UTC

POST /api/v1

```
{
  "currentNode": "",
  "complete": true,
  "context": {},
  "parameters": [],
  "extractedParameters": {},
  "speechResponse": [
    "The rate of 1 bitcoin is <b>$ 21,505.2291</b><br>\nTime : Jun 14, 2022 21:42:00 UTC"
  ],
  "intent": {
    "object_id": "62a82438d5802296cb11568d",
    "confidence": 0.6017876412943343,
    "id": "bitcoin"
  },
  "input": "bitcoin",
  "missingParameters": [],
  "owner": "chat",
  "date": "2022-06-14T21:43:30.985Z"
}
```

Figure 3.4: Test Chat



Train your intent

Train your story for possible user inputs. Tag required parameters using mouse and give em labels

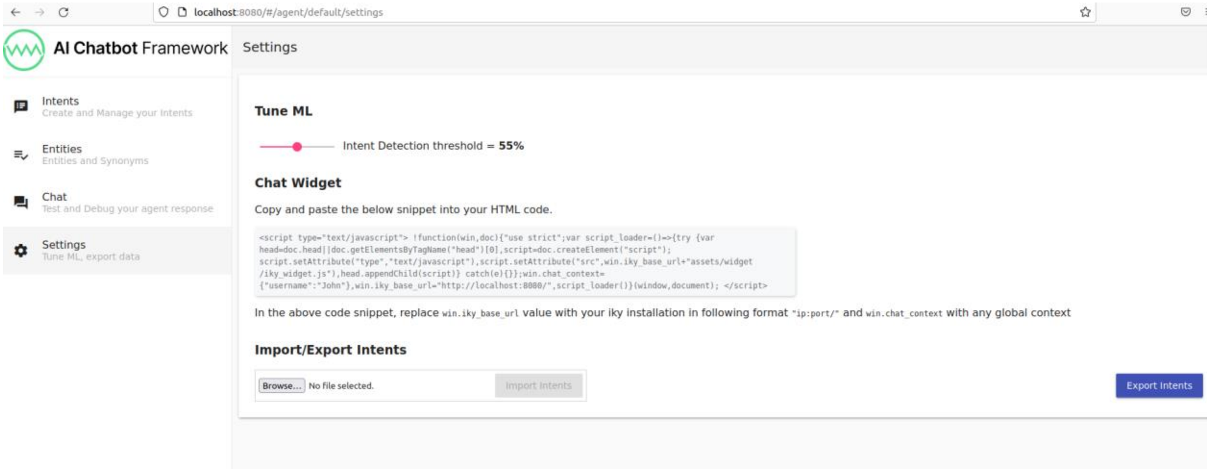
bitcoin market prices

bitcoin market

bitcoin

bitcoin prices

Figure 3.5: training screen



Tune ML

Intent Detection threshold = **55%**

Chat Widget

Copy and paste the below snippet into your HTML code.

```
<script type="text/javascript"> !function(win,doc){"use strict";var script_loader=()=>{try {var
head=doc.head||doc.getElementsByTagName("head")[0],script=doc.createElement("script");
script.setAttribute("type","text/javascript"),script.setAttribute("src",win.iky_base_url+"assets/widget
/iky_widget.js"),head.appendChild(script)} catch(e){}};win.chat_context=
{"username":"John"},win.iky_base_url="http://localhost:8080/",script_loader()}(window,document); </script>
```

In the above code snippet, replace `win.iky_base_url` value with your iky installation in following format `"ip:port/"` and `win.chat_context` with any global context

Import/Export Intents

Figure 3.6: Platform Settings

3.3 Tools used

The tools used for the project:

- Python
- Flask
- Angular
- Docker
- SpaCy

3.3.1 Python

Guido Rossum created Python, an object-oriented programming language, in 1989. It is optimised for rapid prototyping of complex applications.Python is widely used in artificial intelligence, natural language generation, neural networks, and other advanced computer science fields. Python is a potent programming language that can be used to create games, GUIs, and web applications. It is an advanced language. Python is an object-oriented programming language that allows users to create and execute programmes by managing data structures or objects. Everything in Python is in fact of the highest quality. Python treats all objects, data types, functions, methods, and classes equally. Programming languages are developed to meet the needs of programmers and users for an effective tool to create applications that have an impact on lives, lifestyles, the economy, and society. They contribute to the improvement of life by enhancing productivity, communication, and potency. Languages become extinct and obsolete when they fail to live up to expectations and are supplanted by more potent languages. Python is an artificial programming language that has withstood the test of time and remained relevant across industries, businesses, and among individual programmers. It is a living, thriving, and incredibly useful language that is highly recommended as a primary programming language for those who wish to begin programming.

3.3.2 Flask

Flask is a Python-based web application framework. It is created by Armin Ronacher, who leads an international group of Python enthusiasts called Pocco. Flask’s framework is more explicit than Django’s and easier to learn, as it requires less base code to implement a simple web application. Flask relies on the WSGI (Web Server Gateway Interface) toolkit and the Jinja2 template engine. Flask was made to be simple to use and expand. Flask's purpose is to provide a solid foundation for web applications of varying complexity. Flask is useful for a variety of applications. It is particularly useful for prototyping. Flask relies on the following external libraries:the Jinja2 template engine and the Werkzeug WSGI framework.

Flask has a lightweight and modular design, so it is simple to transform it into the required web framework by adding a few modules. It is incredibly simple to deploy Flask in production and it features HTTP request handling and high flexibility. The configuration is even more flexible than Django's, providing a multitude of solutions for all production needs.

3.3.3 Angular

Angular is a development platform, built on TypeScript. As a platform, Angular includes component-based framework for building scalable web applications, collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more, suite of developer tools to help you develop, build, test, and update your code With Angular, taking advantage of a platform that can scale from single-developer projects to enterprise-level applications. Angular is designed to make updating as straightforward as possible, so take advantage of the latest developments with a minimum of effort. Best of all, the Angular ecosystem consists of a diverse group of over 1.7 million developers, library authors, and content creators.

3.3.4 Docker

Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine. It was first started in 2013 and is developed by Docker, Inc.

3.3.5 SpaCy

spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. The library is published under the MIT license and its main developers are Matthew Honnibal and Ines Montani, the founders of the software company Explosion.

3.4 Algorithms

3.4.1 Support vector machine

Support Vector Machine, or SVM, is one of the most popular algorithms for Classification and Regression problems in Supervised Learning. In Machine Learning, it is used primarily for Classification problems. The objective of the SVM algorithm is to generate the optimal line or decision boundary that divides n-dimensional space into classes, so that future data points can be easily classified. This optimal decision boundary is referred to as a hyperplane. SVM selects the extreme points/vectors that contribute to the formation of the hyperplane. These extreme cases are referred to as support vectors, and the corresponding algorithm is known as the Support Vector Machine as shown in Figure 3.7.

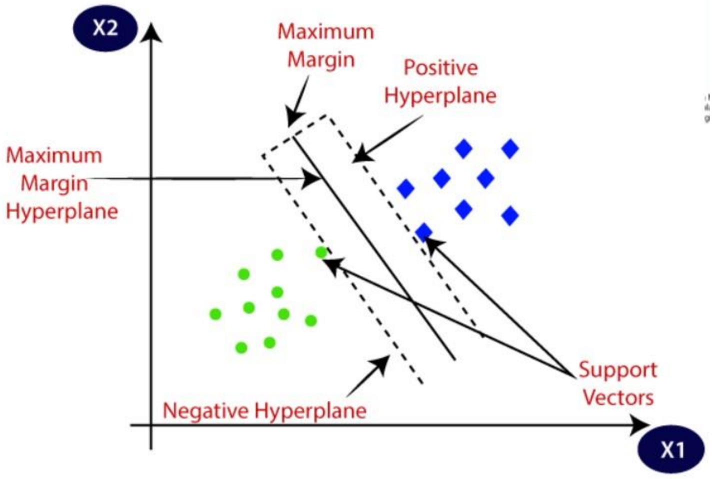


Figure 3.7: support vector machine

3.4.2 Naive Bayes classifier

Naive Bayes classifiers are a collection of Bayes' Theorem-based classification algorithms. It is not a single algorithm, but rather a family of algorithms that share a common principle, namely that each pair of features being classified is independent from the other as shown in figure 3.8

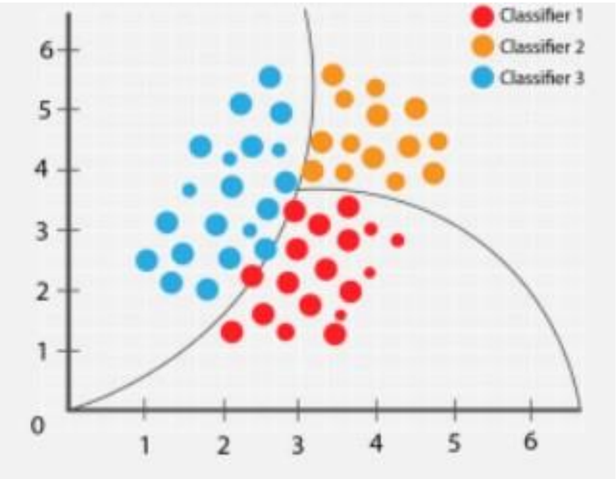


Figure 3.8 Naive Bayes classifier

3.4.3 Natural language processing

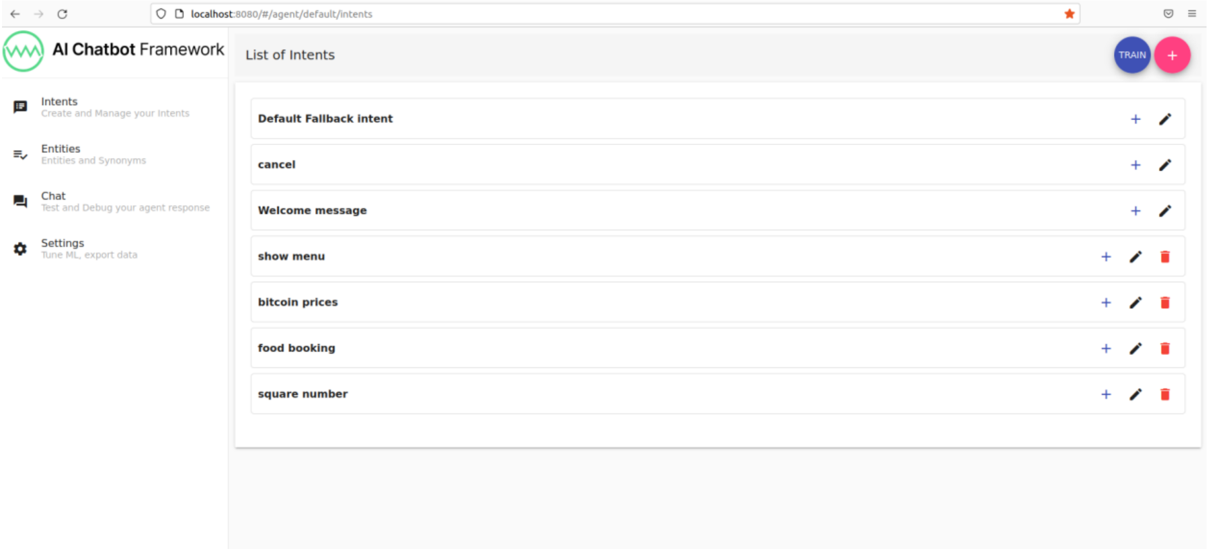
Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

Chapter 4

RESULTS AND DISCUSSIONS

4.1 Graphical user interface

The Main Intents screen is where we can view all the intents in the chatbot, new intents can be created as well as existing ones can be edited as shown in Figure 4.1. Next the detailed intent creation screen as shown in Figure 4.2 where new intents can be created. The test chat area is shown in Figure 4.3 where trained intents can be accessed using chat. The training screen shown in Figure 4.4 helps in training the intent using example chat snippets. Finally Figure 4.5 shows the UI where platform settings such as confidence can be updated.


localhost:8080/#/agent/default/intents
AI Chatbot Framework
Intents
Create and Manage your Intents
Entities
Entities and Synonyms
Chat
Test and Debug your agent response
Settings
Tune ML, export data
TRAIN
+

List of Intents

Default Fallback intent

cancel

Welcome message

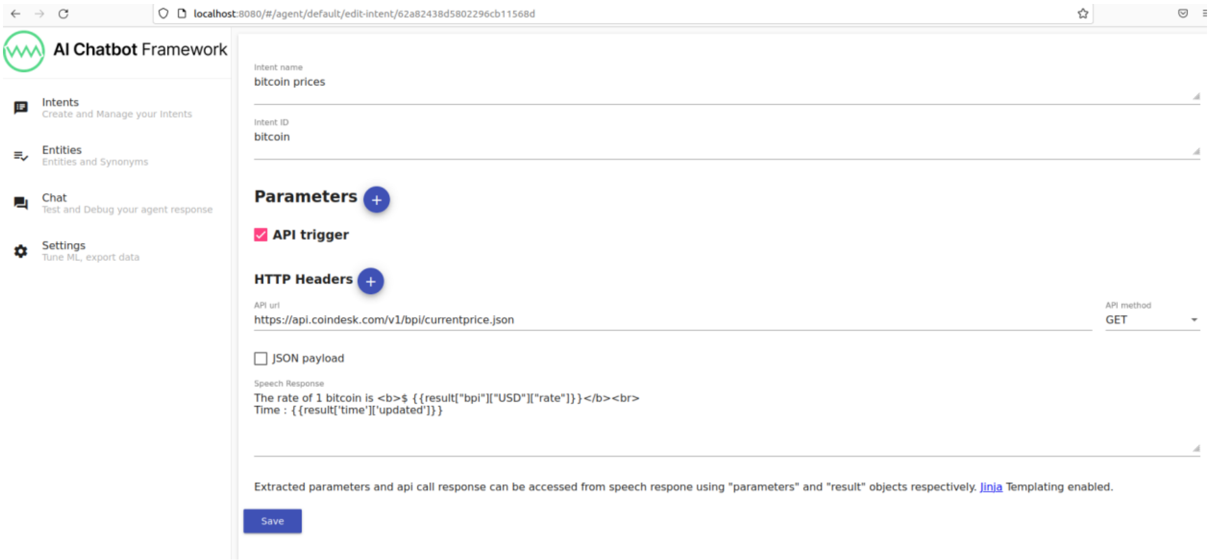
show menu

bitcoin prices

food booking

square number

Figure 4.1: Main Intent Screen


localhost:8080/#/agent/default/edit-intent/62a82438d5802296cb11568d
AI Chatbot Framework
Intents
Create and Manage your Intents
Entities
Entities and Synonyms
Chat
Test and Debug your agent response
Settings
Tune ML, export data
Save

Intent name
bitcoin prices

Intent ID
bitcoin

Parameters +

☑ **API trigger**

HTTP Headers +

API url: https://api.coindesk.com/v1/bpi/currentprice.json

API method: GET

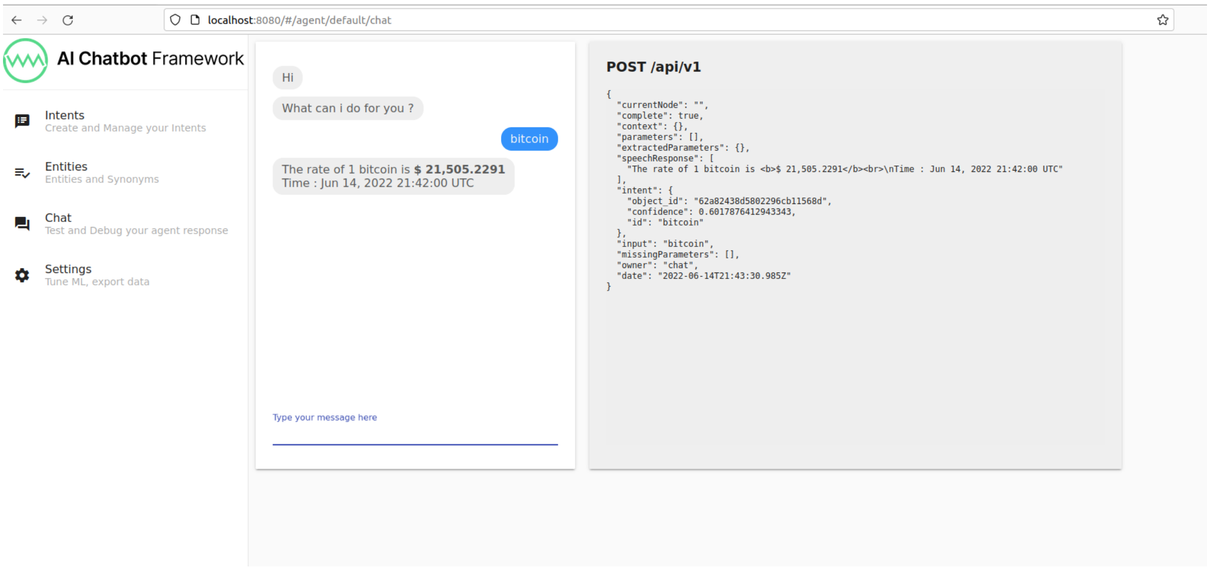
□ JSON payload

Speech Response

```
The rate of 1 bitcoin is <b>$ {{result["bpi"]["USD"]["rate"]}}</b><br>
Time : {{result['time']['updated']}}
```

Extracted parameters and api call response can be accessed from speech respone using "parameters" and "result" objects respectively. Jinja Templating enabled.

Figure 4.2: Intent Creation



Hi

What can i do for you ?

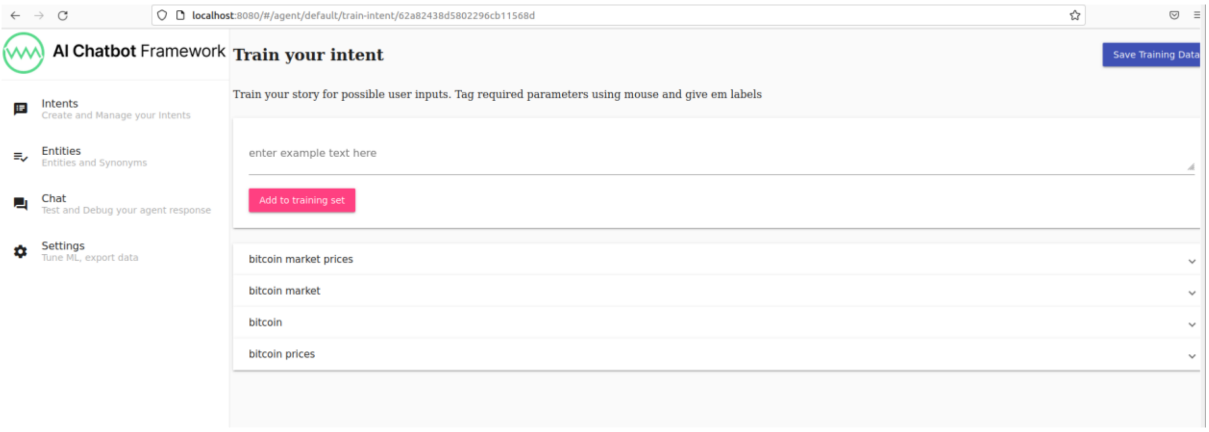
bitcoin

The rate of 1 bitcoin is **$ 21,505.2291**
Time : Jun 14, 2022 21:42:00 UTC

POST /api/v1

```
{
  "currentNode": "",
  "complete": true,
  "context": {},
  "parameters": [],
  "extractedParameters": {},
  "speechResponse": [
    "The rate of 1 bitcoin is <b>$ 21,505.2291</b><br>\nTime : Jun 14, 2022 21:42:00 UTC"
  ],
  "intent": {
    "object_id": "62a82438d5802296cb11568d",
    "confidence": 0.6017876412943343,
    "id": "bitcoin"
  },
  "input": "bitcoin",
  "missingParameters": [],
  "owner": "chat",
  "date": "2022-06-14T21:43:30.985Z"
}
```

Figure 4.3: Test Chat



Train your intent

Train your story for possible user inputs. Tag required parameters using mouse and give em labels

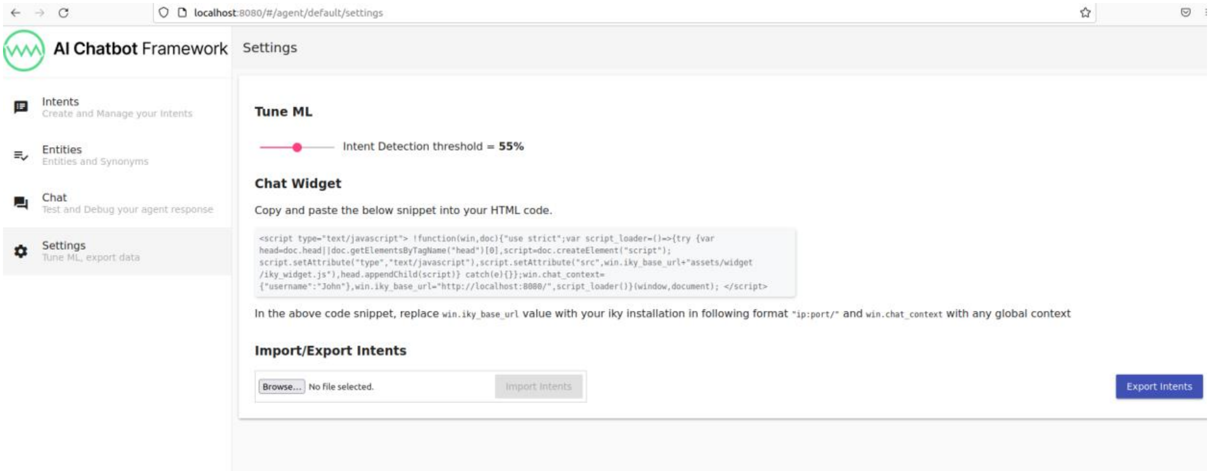
bitcoin market prices

bitcoin market

bitcoin

bitcoin prices

Figure 4.4: training screen



Settings

Tune ML

Intent Detection threshold = **55%**

Chat Widget

Copy and paste the below snippet into your HTML code.

```
<script type="text/javascript"> !function(win,doc){"use strict";var script_loader=()=>{try {var
head=doc.head||doc.getElementsByTagName("head")[0],script=doc.createElement("script");
script.setAttribute("type","text/javascript"),script.setAttribute("src",win.iky_base_url+"assets/widget
/iky_widget.js"),head.appendChild(script)} catch(e){}};win.chat_context=
{"username":"John"},win.iky_base_url="http://localhost:8080/",script_loader()}(window,document); </script>
```

In the above code snippet, replace `win.iky_base_url` value with your iky installation in following format `"ip:port/"` and `win.chat_context` with any global context

Import/Export Intents

Figure 4.5: Platform Settings

Chapter 5

CONCLUSION

The successful completion of this project will yield an application which will be able to create chatbots for small and medium businesses with moderate use cases without any programming and with minimum technical knowledge. This can be further enhanced to be able to create a robust and industry grade product.

Chapter 6

FUTURE SCOPE

The platform can be given more functionality as per user requirement after deployment. Better NLP tools can be integrated into the project to improve its performance.

REFERENCES

[1] G. Daniel, J. Cabot, L. Deruelle and M. Derras, "Xatkit: A Multimodal Low-Code Chatbot Development Framework," in IEEE Access, vol. 8, pp. 15332-15346, 2020, doi: 10.1109/ACCESS.2020.2966919.

[2] G. A. Santos, G. G. de Andrade, G. R. S. Silva, F. C. M. Duarte, J. P. J. D. Costa and R. T. de Sousa, "A Conversation-Driven Approach for Chatbot Management," in IEEE Access, vol. 10, pp. 8474-8486, 2022, doi: 10.1109/ACCESS.2022.3143323.

[3] L. Zhang, Y. Yang, J. Zhou, C. Chen and L. He, "Retrieval-Polished Response Generation for Chatbot," in IEEE Access, vol. 8, pp. 123882-123890, 2020, doi: 10.1109/ACCESS.2020.3004152.