A Report on

Dexter- The College FAQ ChatBot

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

in

Information Technology

by

Ajinkya Huddar (15104047)
Chintan Suchak (16104069)
Chaitanya Bysani (16104045)

Dr. Uttam D. Kolekar
Prof. Kaushiki Upadhyaya

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI
2019-2020

CERTIFICATE

This is to certify that the project Synopsis entitled ***“Dexter - The FAQ Chatbot”*** Submitted by ***“Chintan Suchak (16104069)”” Ajinkya*** **Huddar (15104047)”” Chaitanya Bysani (16104045)”** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology.*** to the University of Mumbai, is a bonafide work carried out during academic year 2019-2020

(Prof. Kaushiki Upadhyaya)	(Dr Uttam D.Kolekar)
Co-Guide	Guide

Prof. Kiran Deshpande	Dr. Uttam D.Kolekar
Head Department of Information Technology	Principal

External Examiner(s)

1.

2.

Place: A.P. Shah Institute of Technology,
Thane

Date:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

—————————————————

(Ajinkya Huddar 15104047)

—————————————————

(Chintan Suchak 16104069)

—————————————————

(Chaitanya Bysani 16104045)

Date:

Abstract

Chatbot is a new and upcoming technology which has great demand in various industries. The main goal of a chatbot is to create a human like conversation between a human and a machine, so as to reduce the work stress. Chatbot can be implemented in any industry easily, unlike any other product where the products needs to be developed and tested before switching platform. In colleges, especially during the time of admission, reception gets crowded and people have to wait to get their queries solved. If any person wants to know about the college, then he/she has to travel all the way to the college. Al though every college has its own website, not everybody is able to find the answer to their query. Colleges are not working on weekends, so if someone wants to visit or call reception to get their query answered they will have to wait until any working weekday. To solve these problems, we will create a AI chatbot. This chatbot will be embedded on the college website and will be able to answer any college-related query easily. Chatbot will be able to answer multiple persons at the same time, people don't have to visit the college to get their query solved and it will be available 24/7.

Contents

List of Figures

List of Abbreviations

YAML:	Yet Another Markup Language
NLU :	Natural Language Understanding
MD:	MARKDOWN file
JSON:	JavaScript Object Notation
RNN:	Recurrent Neural Network
LSTM:	Long Short Term Memory

Chapter 1

Introduction

A Chatbot is an Artificial Intelligence (AI) software program that conducts a conversation via auditory or textual methods. Such programs are often created to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chatbots are mainly used in dialog systems for various practical purposes including customer services or information acquisition. The chatbot is a technology that is growing fast. Chatbots are being used increasingly in many other sectors, such as banking, entertainment, news, customer services as well as in the medical sector. In this project, we will be developing one such chatbot which will help solve any queries which are associated with college FAQs. Whenever a student takes admission in a new college or wants to take admission in a new college, they might have lots of queries in their mind.

Students might be reluctant to ask about the query to any faculty or reception andmight end up assuming the wrong answer or it might be the case where a student is not able to take out time from his/her busy schedule. Developing a chatbot solves the issues that may arouse in gathering needed data. It will be accessed from any place at any time. It would give user-friendly interaction to the users. One such good example of chatbot will be paper [6] and [10], where the authors have created chatbots for solving queries of related to a specific topic. In college websites, most users aren't able to realize the desired data which in turn makes the website pointless. This issue can be solved with the help of chatbots So to solve all these difficulties, we have proposed an AI Chatbot. Idea for this chatbot was drawn out of paper [7], where the authors have created a chatbot for solving queries related tocollege. Artificial Intelligence is a technology that provides human-like intelligence to a machine. This chatbot will be using this technology to create an answer for even those queries whose answers are not available in the database.

Rasa comes up with 2 components —

Rasa NLU — a *library* for natural language understanding (NLU) which does the classification of intent and extract the entity from the user inpRASA stack is an open-source AI tool and being an opensource framework, It is easy to customize. In fact, In many cases, Clients do not want to share their data and the majority of the tools available are cloud-based and provide software as a service. You can not run them internally in your environment. So you need to send your data to the third party. With RASA, There is no such issue. You can build, deploy or host Rasa internally in your server or environment with complete control on it and helps bot to understand what the user is saying.

Rasa Core — a chatbot *framework* with machine learning-based dialogue management which takes the structured input from the NLU and predicts the next best action using a probabilistic model like LSTM neural network.

NLU and Core are independent and one can use NLU without Core, and vice versa. Though Rasa recommends using both.

Chapter 2

Literature Review

The papers referred are mentioned below:

[1] Nudtaporn Rosruen and Taweesak Samanchuen "Chatbot Utilization for Medical Consul- tant System", The 2018 Technology Innovation Management and Engineering Science Inter- national Conference(TIMES-iCON2018)

The authors have described about how the number of individuals looking for well-being data from the web increments drastically. A few elements impact individuals to utilize the web for scanning for well-being data. Confided in restorative data, for exam- ple, infections, side effects, and treatment is important for individuals to deal with some broad sickness or being utilized as a bit of choice help data before visiting a specialist. In this work, the therapeutic guide framework called" MedBot" was created by utilizing Dialog flow controlled by Google's machine learning. The learning base for correspondence comprise of 16 symptoms. The chatbot can be executed in Instant Messaging (IM) application, or online for example, Facebook, Hangout, and Line by utilizing the given API's. In this work, Line is utilized as the test framework for examination. The target of this work is to expand the administration capacity and decline the activity cost of medical consultancy administration by utilizing the chatbot. The down side to this chatbot was, there was no implementation of the neural network making it a chatbot which will only answer those questions which are fed in the chatbot's database. This restricted the knowledge base of the chatbot to the minimum value and to in- crease the knowledge base values had to be fed in the database manually.

[2] SathitPrasomphan, "Improvement of Chatbot in Trading System for SMEs by Using Deep Neural Network",2019IEEE4th International Conference on Cloud Computing and Big Data Analytics.

The author describes about a chatbot whose exploration displays a strategy for creating chatbots to serve their clients. All in all, these chatbots are utilized for responding to inquiries in numerous organizations, giving client data, giving train plans ,making a difference client reservations, menial helpers; fill in as call focuses to serve ten million clients naturally. A profound learning based conversational man-made brain power procedure was utilized as apparatuses for learning discussion among machine and client. Additionally, the means required are the procedure utilized related to the convolution neural system strategy by utilizing Tensor- flow preparing to improve the precision of these chatbots. From the exploratory out comes, utilizing profound learning for chatbots learning, the precision is superior to the customary model. This article has not used cloud for their databases to rage which takes a little longer to fetch data.Hence the speed off etching is slow.

[3] Panitan Muangkammuen, Narong Intiruk, Kanda Runapongsa Saikaew, "Automated Thai- FAQ Chatbot using RNN-LSTM"

The author describes how the quantity of web-based business clients has expanded quickly.In 2017,the number of computerized purchasers was over 1.66 billion individ- uals world wide up from 1.32 billion in 2014. The paper also gives information about on line shops which regularly require administrations, for example, live talk for clients support. In any case, such live talk needs administrators to hold on to visit with clients. Then again,if shops give online clients that administrators work for just certain hours, at that point clients, need to sit tight for a response for quite a while. As the quantity of clients has expanded by about 10 percent every year, the interest for the client the administration additionally increments.Great online client administrations will prompt higher consumer loyalty and developing benefit.To tackle the problem of solving queries related to e-commerce, authors came up with an idea of converting online client support methodologies to a robotized chatbot to reply to clients' inquiries consequently. The chatbot will deal with client issue reports and answers about sim- ilar answers for a similar sort of issues. One of the ideal and effective online client support methodologies are to give a robotized chatbot to reply to clients' inquiries consequently.The chatbot will deal with client issue reports and answers about similar answers for a similar sort of issues. This article researched how to structure and build up a chatbot to reply FAQs in a particular space. Specifically, they utilized a profound learning AI model that was fit for gaining from a enormous information and further more utilized LSTM for managing a arrangement in language. The AI model will order expressions of inquiries with each class having their answers. The downside to this article was, the chatbot developed was retrieval- based chatbot, which means it will not be able to answer anything outside its database.

[4] Rupesh Singh, Harshkumar Patel, Manmath Paste, Nitin Mishra, Nirmala Shinde, "Chat- bot using TensorFlow for small Businesses", Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant - Part Number: CFP18BAC-ART; ISBN:978- 1-5386-1974-2.

The authors describe chatbots as a software program used in the entertain- ment industry, agencies and user support. Chatbots are modeled on various strategies such as knowledge base, machine learning-based. Machine learning primarily based on chatbots yield greater practical results. Chatbot which gives responses primarily based on the context of con- versation tends to be extra consumer - friendly. The chatbot they are proposing demonstrates a technique of creating chatbot which can follow the context of the conversation. This method uses Tensorflow for growing the neural network model of the chatbot and uses the NLP meth- ods to keep the context of the conversation. These chatbots can be used in small industries or businesses for automating customer care, as user queries will be handled through chatbots, therefore, decreasing want of human labor and expenditure. The authors have created a chat- bot which can be used as a reference for creating chatbots for various purpose. This chatbot will be having limited information about the purpose and will provide data to the user from its database. The database will have to be updated manually if there are any changes in the future.

[5] Rohit Binu Mathew, Sandra Varghese, Sera Elsa Joy, Swanthana Susan Alex, "Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning", Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32- ART; ISBN: 978-1-5386-9439-8

Authors areemphasizing onthe health ofa person, about how people nowadays are concerned more about their work and not their health. To save the time of visiting doctor for regular checkups, authors havecreatedachatbot which helps in predicting the diseasesfrom the symptoms provided from the user. It also provides a link for the treatment of identified diseases. It is an android based chatbot. To accomplish their goal, they have used a machine learningalgorithm, they haveused K- nearest neighbor algorithm (KNN). Thisalgorithm maps the symptoms provided by the user with the predefined dataset.

Chapter 3

Existing Architecture

Here in the figure shown below, there is a Web Interface between the user and the chatbot where in the user input the query in text format and from there it goes to the Chatbot which consists of Reasoning Component and Knowledge Base. In Reasoning Component the chatbot checks whether the input of the query is matching with the entities and actions and if not it goes to the Knowledge base which consists of the group of queries output. So it matches with the input of the query and gives the suitable output in the form of response.

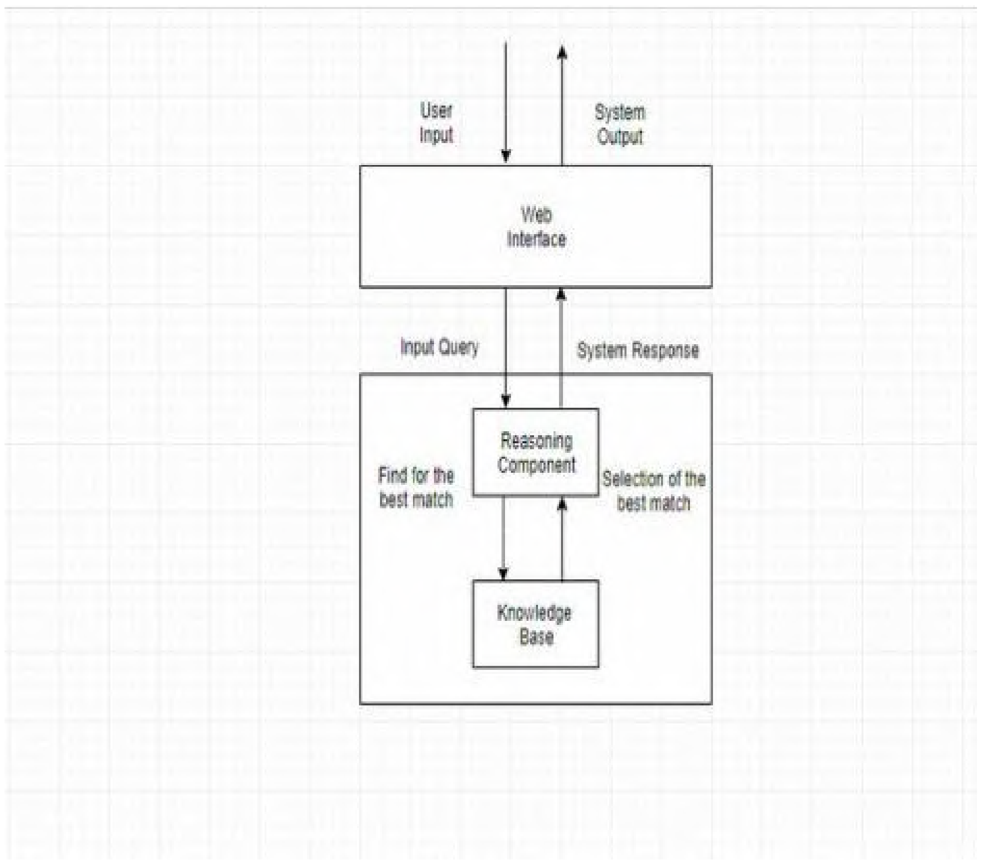


Figure 3.1: Existing System Architecture

We will presently talk about the engineering of existing chatbot frameworks accessible with their applications. This chatbot bot is notable and has won numerous honors also, the chatbot is called ALICE.

ALICE(Artificial Linguistic Internet Computer Entity) is an honor winning open-source natural language artificial intelligence chatbot which uses AIML(Artificial Intelligence Mark-up Language) to shape reactions to questions. It is enlivened by ELIZA and an open-source chatbot created by Dr. Wallace, which depends on common language comprehension and example coordinating. It has won Loebner prize multiple times. It creates reactions to the client's question by applying some example coordinating principles. Nonetheless, it can't finish the Turing assessment, as even the easygoing client will regularly uncover its blemishes in short discussions. The design of chatbot comprises of two isolated parts in particular "chatbot engine" and "language model" which offers us the chance to effectively execute a chatbot in a recently created knowledge model. The language model is put away in AIML files. The essential structure highlight of AIML is moderation and from all the talk robot dialects, AIML is maybe the least complex. As talked about before, the essential unit of information in AIML is the category. Every category comprises information or question, a yield or an answer and a discretionary setting. The inquiry is known as an example. The appropriate response or reaction is the template. The two sorts of the discretionary setting are classified "that" and "topic".

Chapter 4

Objectives

- To reduces the work stress of reception.
- To make navigation through the college website easier for students
- To provide answers regarding “What do I do know?” questions.
- To improve Query Handling System since chatbot is available 24 x 7.
- To help new student get familiar with the new Environment.
- To help students give a brief idea about college so that they can decide whether they should visit the college or not

Chapter 5

Problem Statement

The main reason behind choosing this topic as the project was that many students were facing issues regarding the updates of revaluation examinations or results, about any important notice and events going on in the college. It becomes really difficult for students who stay far away from the college and they just have to come to college for inquiry purpose. Even reception becomes complete chaos during the time of admission, many students and parents visit the college reception to get their queries solved. The receptionist will only be able to handle 2To 3 person at a time and others will have to wait for their turn. This will also cause tiredness for receptionist. To overcome this problems, we are making the graphical user interface inquiry chatbot which gives 24*7 updates regarding any ongoing events or notice. The main motive is to design a chatbot which will simulate a conversation with any user and provide them suitable answers regarding any college-related queries.

Chapter 6

UML Diagram

Activity Diagram:

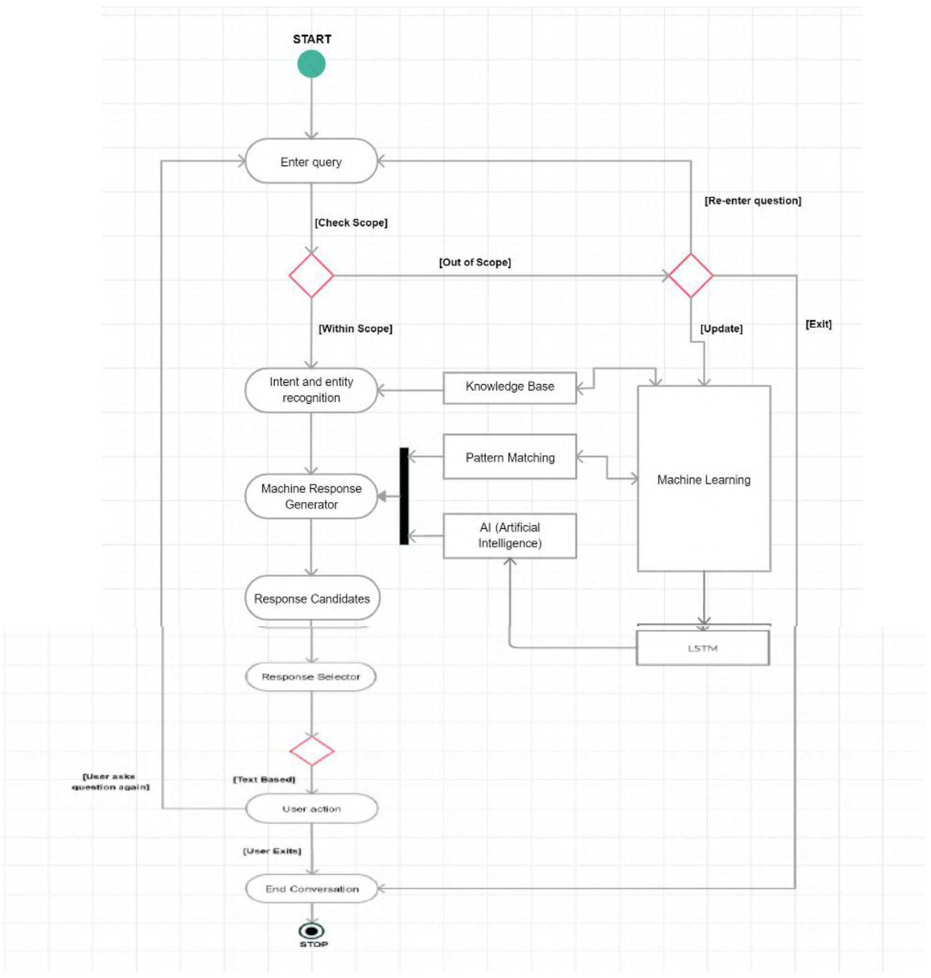


Figure 6.1: Activity Diagram

The user first input his query in the form of text, when the query is sent it first checks whether the query is in scope or out of scope. If the query is in scope then the intent and entity recognition takes place and then the Machine Response Generator forms the statement depending upon the user query the response candidate and response selector is generated. If the question or query is out of scope then it gets generated in Machine Learning and LSTM where the query is generated and updated and sent to Knowledge Base, Pattern Matching and AI. If the user keeps asking repeated set of question and query then it automatically gets updated in User action.The NLU keeps on updating and training the model for efficient and better query output.

Class Diagram:

The different entities in this environment are:

1. Guest - The various number of users.

2. Enquiry- Students just have to query through the bot which is used for chating.

3. Student- a person who is studying at a university or other place of higher education.

4. Faculty- a group of university departments concerned with a major division of knowledge.

5. Login- A login is a set of credentials used to authenticate a user in Chatbot

6. Administrator- a person responsible for carrying out the administration of a Chatbot.

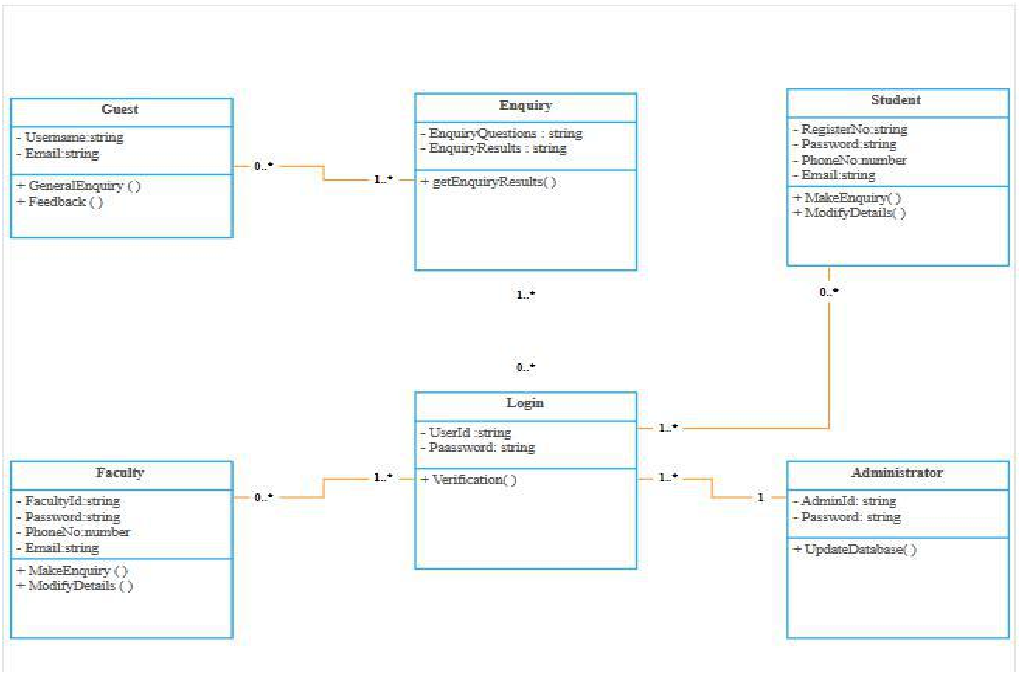


Figure 6.2: Class Diagram

Chapter 7

Project Design

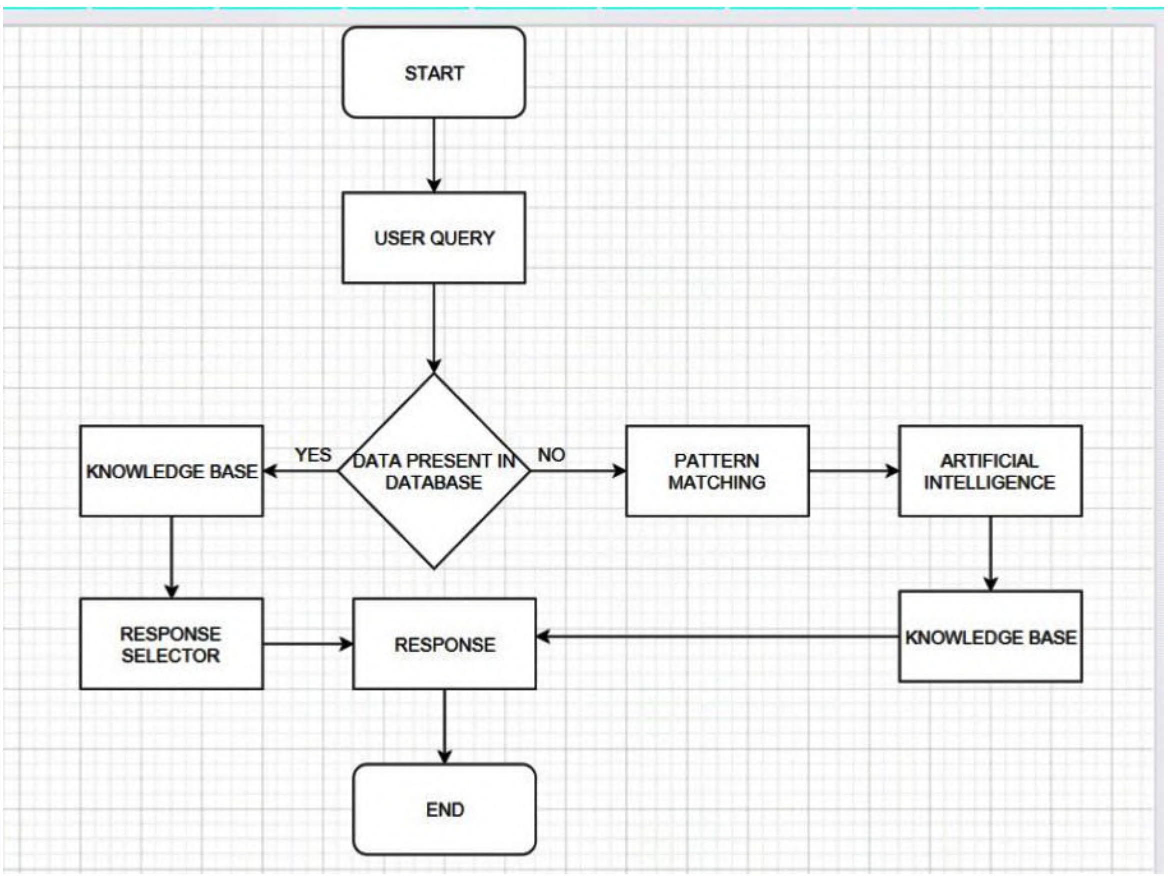


Figure 7.1: Project Design(Flow Chart)

The Project Design comprises of the following:

1. User Query: It consists of the user input or question in the form of text query.

2. Knowledge Base : A knowledge base is a technology used to store complex structured and unstructured information used by a computer system. Its is more like a database.

3. Response Selector: The response selector just scores all the response candidate and selects a response which should work better for the user.

4. Pattern Matching: It is used to classify text and produce a suitable response for users or students.

5. Artificial Intelligence: The chatbot will have Artificial Intelligence which will allow it to produce answers for eventhose question whose answers arenot defined. Artificial Intelligence will be provided with the help of machine learning. Machine Learning will be done in Python language. The chatbot will be Generative model because of which it will be able to generate new responses from scratch. It will be done by using RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory).

6. Response: It is more like a output given to the user depending upon the query of the user.

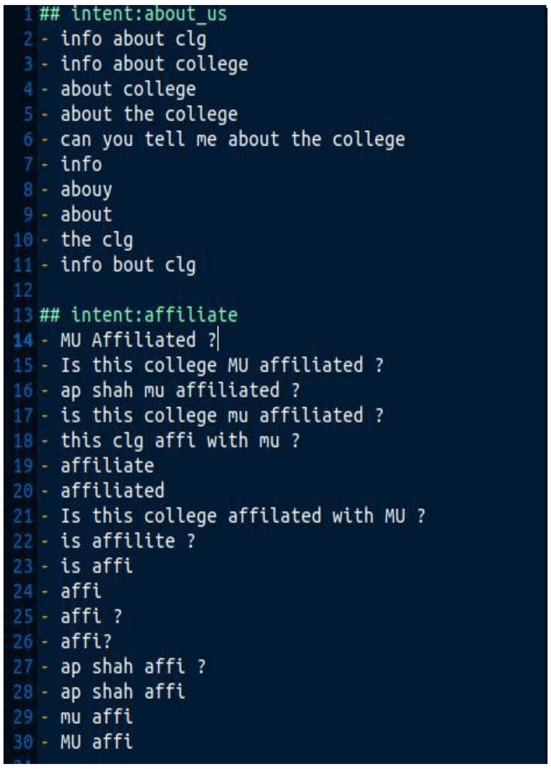
Chapter 8

Project Implementation

Working of the Chatbot: User will first ask their query to chatbot, Chatbot will check if the answer to that query is available in database or not. If the answer to the query is available, it will provide respective answer If the answer is not present, it will perform pattern matching and artificial intelligence to build an answer and will send an alert to admin to add this query to the database. After that it will provide the response and ask if the user has any more query If not the chatbot closes.

nlu.md file -
The first piece of a Rasa assistant is an NLU model. NLU stands for Natural Language Understanding, which means turning user messages into structured data. To do this with Rasa, you provide training examples that show how Rasa should understand user messages, and then train a model by showing it those examples.

Intent —
Intent is nothing but what the user is aiming for. For example — if the user says *“Reserve a table at Cliff House tonight”* the intent can be classified as to book the table.



```
## intent:about_us
- info about clg
- info about college
- about college
- about the college
- can you tell me about the college
- info
- abouy
- about
- the clg
- info bout clg

## intent:affiliate
- MU Affiliated ?
- Is this college MU affiliated ?
- ap shah mu affiliated ?
- is this college mu affiliated ?
- this clg affi with mu ?
- affiliate
- affiliated
- Is this college affilated with MU ?
- is affilite ?
- is affi
- affi
- affi ?
- affi?
- ap shah affi ?
- ap shah affi
- mu affi
- MU affi
```

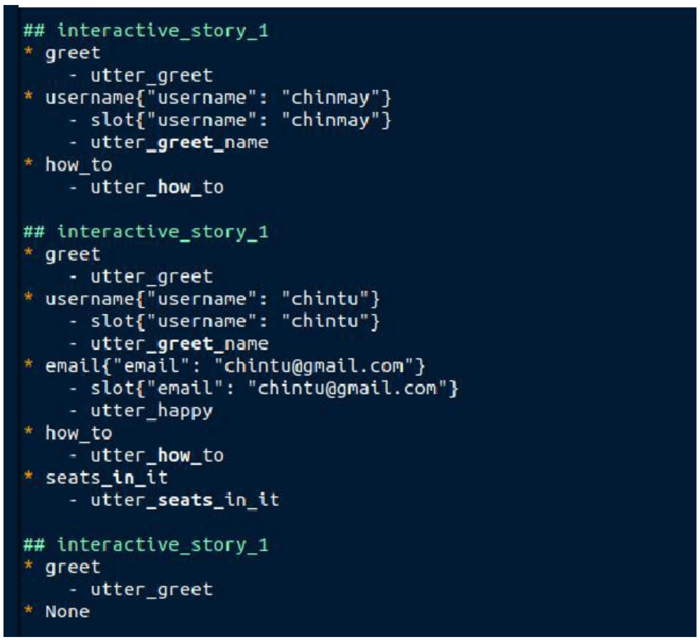
Figure 8.1: Intent

Stories.md file -

Rasa stories are a form of training data used to train the Rasa's dialogue management models. A story is a representation of a conversation between a user and an AI assistant, converted into a specific format where user inputs are expressed as corresponding intents (and entities where necessary) while the responses of an assistant are expressed as corresponding action names. A training example for the Rasa Core dialogue system is called a story.

Interactions:

A conversation between a user and an AI assistant, converted into a specific format where user inputs are expressed as corresponding intents (and entities where necessary) while the responses of an assistant are expressed as corresponding action names.

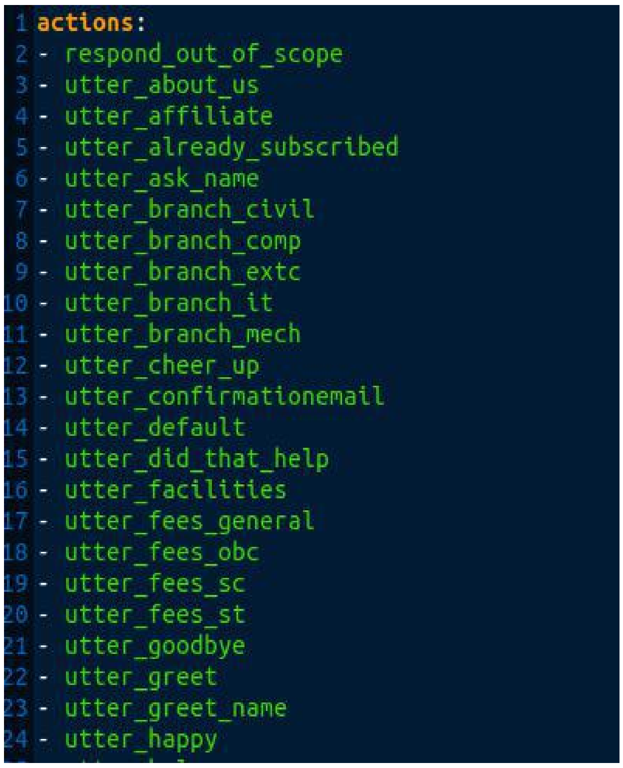


```
## interactive_story_1
* greet
    - utter_greet
* username{"username": "chinmay"}
    - slot{"username": "chinmay"}
    - utter_greet_name
* how_to
    - utter_how_to

## interactive_story_1
* greet
    - utter_greet
* username{"username": "chintu"}
    - slot{"username": "chintu"}
    - utter_greet_name
* email{"email": "chintu@gmail.com"}
    - slot{"email": "chintu@gmail.com"}
    - utter_happy
* how_to
    - utter_how_to
* seats_in_it
    - utter_seats_in_it

## interactive_story_1
* greet
    - utter_greet
* None
```

Figure 8.2: Interactions

Domain.yml -
The domain is the world of our chatbot. It contains everything the chatbot should know, including: All the actions it is capable of doing, the intents it should understand, the template of all the utterances it should tell the user, and much more.

Actions —
Actions are basically the operations performed by the bot either asking for some more details to get all the entities or integrating with some APIs or querying the database to get/save some information.

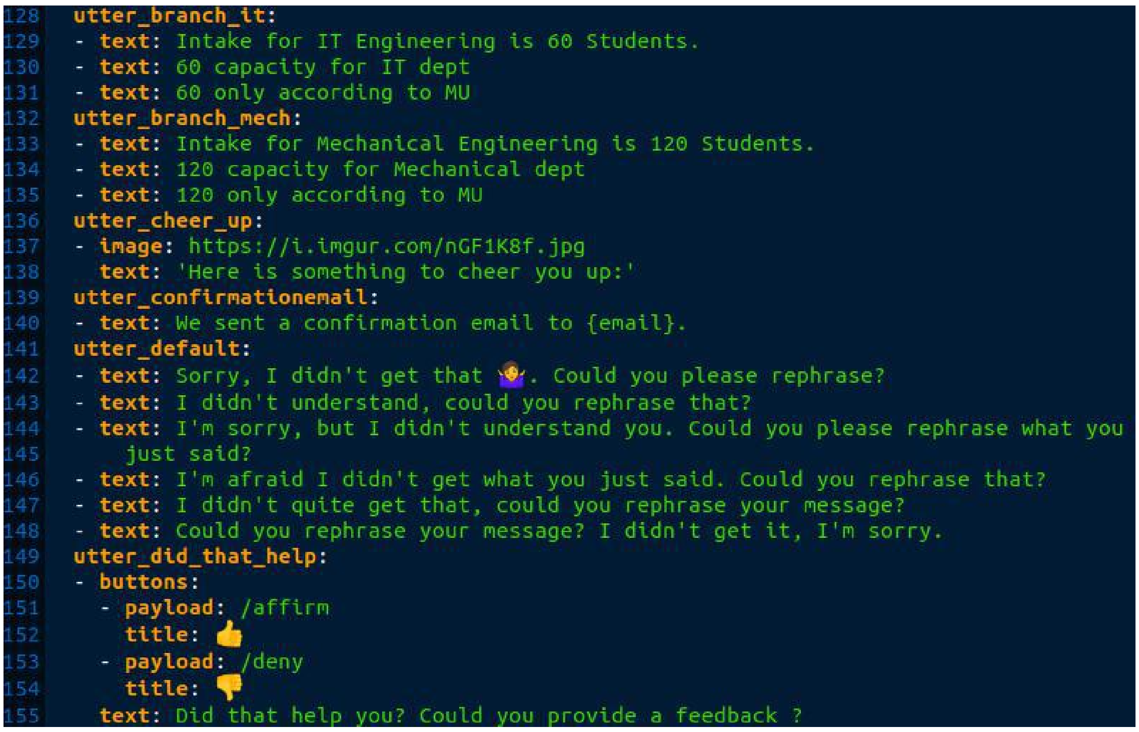


```
actions:
- respond_out_of_scope
- utter_about_us
- utter_affiliate
- utter_already_subscribed
- utter_ask_name
- utter_branch_civil
- utter_branch_comp
- utter_branch_extc
- utter_branch_it
- utter_branch_mech
- utter_cheer_up
- utter_confirmationemail
- utter_default
- utter_did_that_help
- utter_facilities
- utter_fees_general
- utter_fees_obc
- utter_fees_sc
- utter_fees_st
- utter_goodbye
- utter_greet
- utter_greet_name
- utter_happy
```

Figure 8.3: Action

Entity —

Entity is to extract the useful information from the user input. For example above “*Reserve a table at Cliff House tonight*” the entities extracted would be place and time. Place — Cliff House and Time — tonight



```
utter_branch_it:
- text: Intake for IT Engineering is 60 Students.
- text: 60 capacity for IT dept
- text: 60 only according to MU
utter_branch_mech:
- text: Intake for Mechanical Engineering is 120 Students.
- text: 120 capacity for Mechanical dept
- text: 120 only according to MU
utter_cheer_up:
- image: https://i.imgur.com/nGF1K8f.jpg
  text: 'Here is something to cheer you up:'
utter_confirmationemail:
- text: We sent a confirmation email to {email}.
utter_default:
- text: Sorry, I didn't get that 🤷. Could you please rephrase?
- text: I didn't understand, could you rephrase that?
- text: I'm sorry, but I didn't understand you. Could you please rephrase what you
    just said?
- text: I'm afraid I didn't get what you just said. Could you rephrase that?
- text: I didn't quite get that, could you rephrase your message?
- text: Could you rephrase your message? I didn't get it, I'm sorry.
utter_did_that_help:
- buttons:
  - payload: /affirm
    title: 👍
  - payload: /deny
    title: 👎
  text: Did that help you? Could you provide a feedback ?
```

Figure 8.4: Entity

Visualization of the Model:

The histogram that the script produces allows you to visualise the confidence distribution for all predictions, with the volume of correct and incorrect predictions being displayed by blue and red bars respectively. Improving the quality of your training data will move the blue histogram bars to the right and the red histogram bars to the left of the plot.

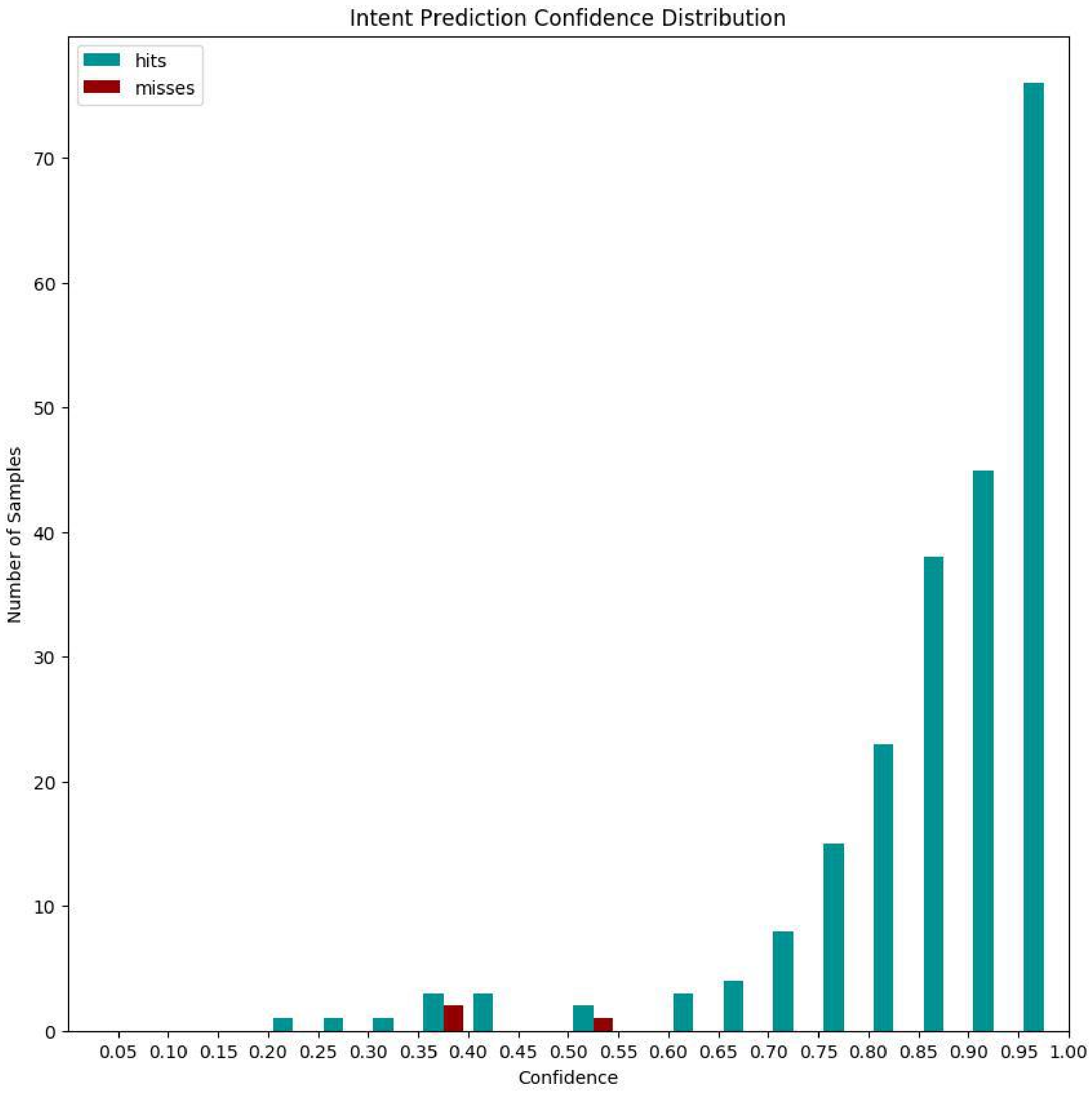


Figure 8.7: Histogram of Confidence V/S Samples

Confusion Matrix:

The confusion matrix shows you which intents are mistaken for others; any samples which have been incorrectly predicted are logged and saved to a file called errors.json for easier debugging.

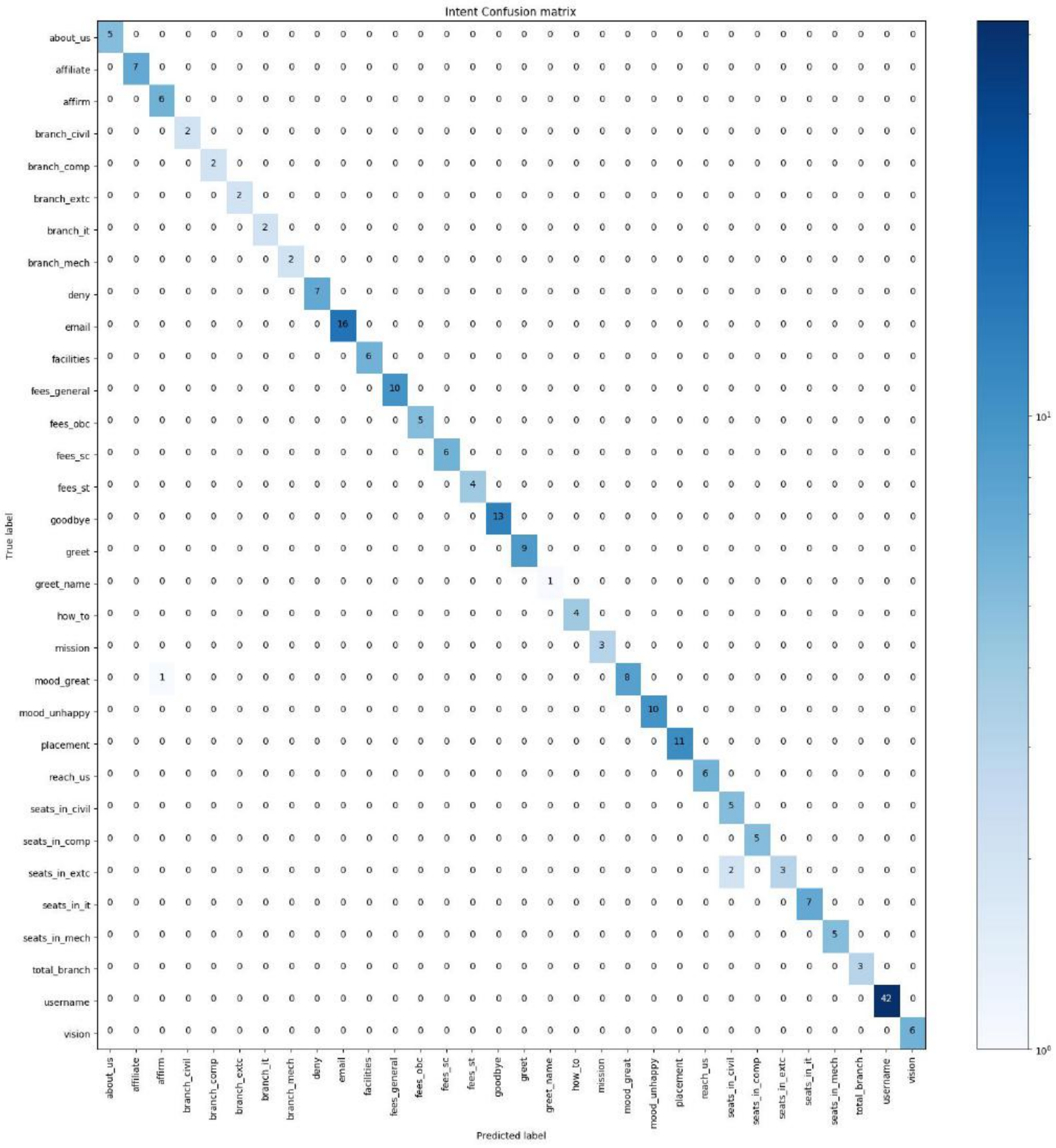


Figure 8.8: Confusion Matrix

Chapter 9

Testing

Before we cover tests that are specific to Rasa and machine learning, let's first take a broader look at testing in software development of Chatbot.

Types of Tests

Software tests follow a hierarchy, moving from granular tests that assess small pieces of code, to higher level tests that evaluate how the entire system works together.

Unit tests

Unit tests evaluate the smallest and most specific pieces of code, usually individual functions or methods. As an example, imagine a function that calls an API to get the latest currency conversion rates and then converts an amount in Euros to USD.

Integration tests

Integration tests operate at a higher level than unit tests, by evaluating how parts of the application work together. However, they don't test the entire application. An integration test typically tests just one feature set or workflow, for example, whether a user can log in to their account.

Functional tests

Functional tests sit one layer higher than integration tests. Functional testing checks how the entire application performs, with all of the pieces working together. Automated functional tests are often run by simulation software because they involve testing the GUI. For example, those who build web-based applications might be familiar with tools like Selenium, which automates actions in the browser.

Tests in Rasa

We've talked about tests in the traditional software sense, but next we'll discuss tests that are specific to Rasa. Instead of testing whether a function returns the expected output or whether a UI is glitchy, these tests measure whether the machine learning models are making correct predictions.

A quick note on terminology: In machine learning, we have the concept of a test set, a portion of labeled data held aside from training to measure the accuracy of the model's predictions.

This technique isn't related to software testing, in and of itself. We'll discuss test sets in greater detail when we cover testing the NLU model.

The Rasa tests we'll discuss in the remainder of this blog post can be run as part of an automated CI/CD pipeline, and they measure an important aspect of your assistant's performance: how accurately the machine learning models are classifying the user's message and predicting the bot's next action. We'll also cover checks you can automate to validate the format of your training data files.

For complete test coverage across your entire application, you'll want to explore other types of tests as well. Custom actions can be tested using unit and integration tests (find more resources on writing tests in Python. For functional testing, you can check out simulators like Botium, which is designed specifically for testing chatbots and voice assistants.

Data Validation

Before you measure the performance of your models, you'll want to be sure the data you're using to train is free of errors. To check your training data for errors and conflicts, run the command:

```
rasa data validate stories --max-history 5
```

This surfaces errors in your training data files, like a training example that appears under more than one intent. Adding the `stories` positional argument and the `--max-history 5` flag runs the story structure validation tool in addition to checking the NLU, domain, and story data files.

The story structure validation tool checks for conflicting stories in your training data. Stories are considered to be in conflict when they have the same conversation history (going back the *max history* number of conversation turns) but specify different bot responses. When this happens, Rasa can't learn the correct next action to take.

NLU

The NLU test command evaluates the NLU model's ability to extract entities and correctly classify intents. There are 2 methods you can use to test your NLU model: splitting your data into training and test sets, and using cross-validation.

Splitting a test set

A test set holds back a portion of training data when training the model. The labeled data that's set aside is used as a benchmark to measure how well the model generalizes in comparison. The data in the test set is selected so it's a representative sample, that is, contains the same proportion of intents as the data used to train the model.

To split your data into a test set, run this command:

```
rasa data split nlu
```

Cross Validation

Cross validation is another technique for testing how well the NLU model generalizes. When you use the method described in the previous section to split a test set, a portion of your training data is never used to train the model (because it's reserved for testing after the model has been trained). Cross validation trains (and tests) the model on your entire data set.

Run a cross validation test with this command:

```
rasa test nlu --cross-validation
```

End-to-End

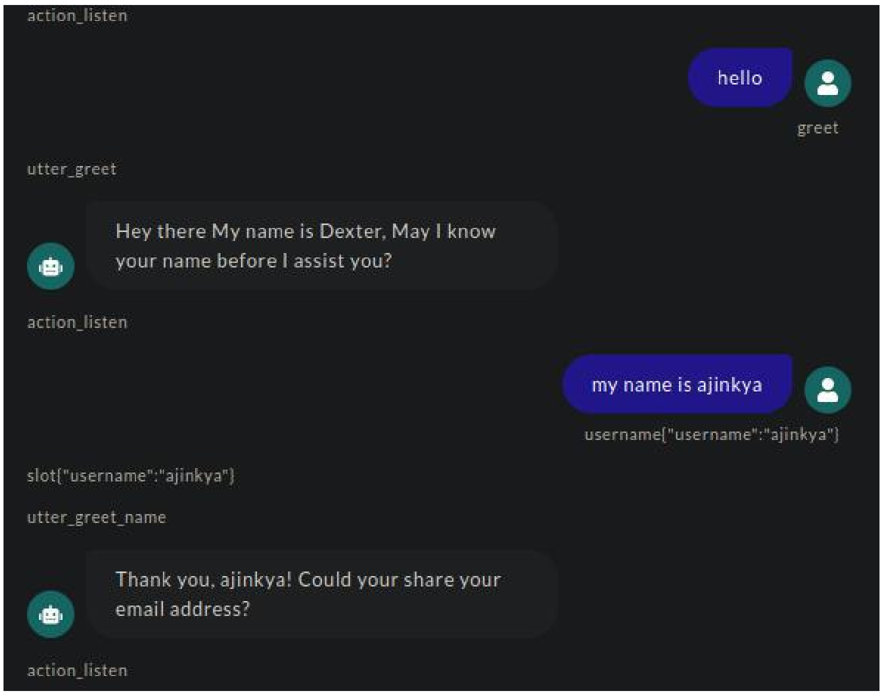
End-to-end tests get their name because they measure how well the models generalize on the entire conversation. Using a test data format called end-to-end stories, this method evaluates both the NLU model and the dialogue management model.

A proper end-to-end test set should represent the kinds of conversations your assistant actually encounters. Including an exhaustive set containing every possible NLU example or dialogue turn is less important than making sure your test set covers the most common types of conversations your assistant encounters in real life. This ensures your assistant behaves in a predictable manner for the types of user flows you intend to support. For that reason, we recommend using Rasa X to generate test stories from real conversations.

Note that unlike the story format used for training, end-to-end test stories also include the full text of the user's message.

Following below is the end to end interaction between the User and the bot.

Image 1: The user is greeting to the bot and in return the bot is asking name of the user.


action_listen

hello

greet

utter_greet

Hey there My name is Dexter, May I know your name before I assist you?

action_listen

my name is ajinkya

username{"username":"ajinkya"}

slot{"username":"ajinkya"}

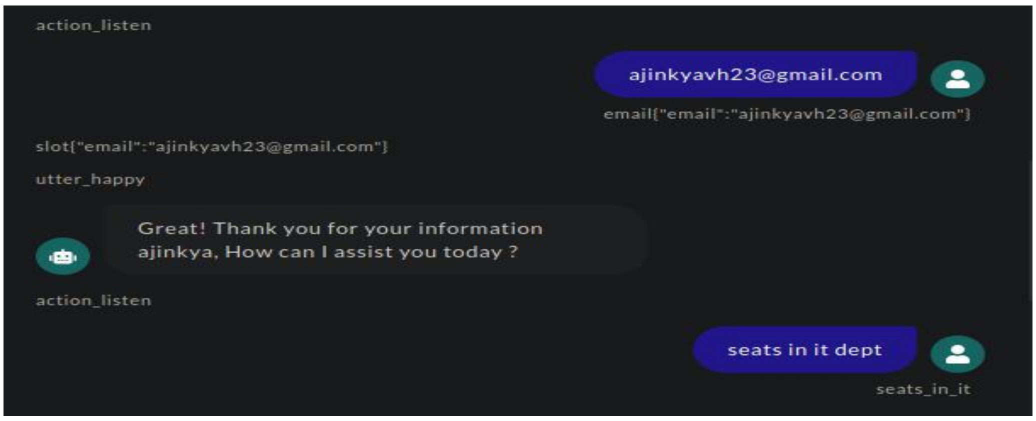
utter_greet_name

Thank you, ajinkya! Could your share your email address?

action_listen

Figure 9.1: Conversation between the user and the bot

Image 2: Where the user provides his name and in return the bot asks his email id


action_listen

ajinkyavh23@gmail.com

email{"email":"ajinkyavh23@gmail.com"}

slot{"email":"ajinkyavh23@gmail.com"}

utter_happy

Great! Thank you for your information ajinkya, How can I assist you today ?

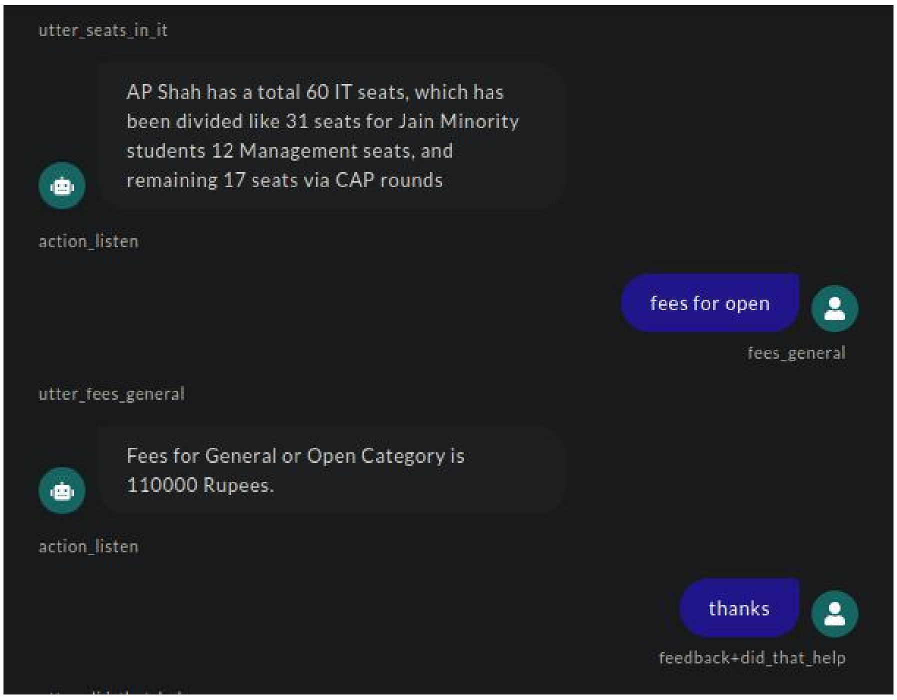
action_listen

seats in it dept

seats_in_it

Figure 9.2: Conversation between the user and the bot

Image 3: Where the user is asking the basic query about the College.



utter_seats_in_it

AP Shah has a total 60 IT seats, which has been divided like 31 seats for Jain Minority students 12 Management seats, and remaining 17 seats via CAP rounds

action_listen

fees for open

fees_general

utter_fees_general

Fees for General or Open Category is 110000 Rupees.

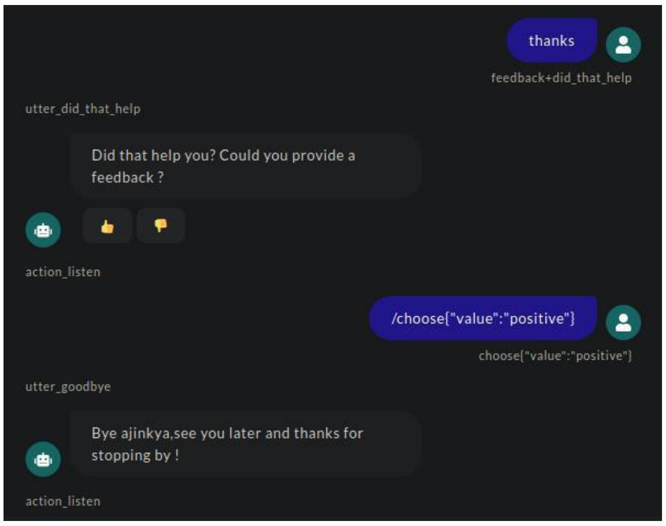
action_listen

thanks

feedback+did_that_help

Figure 9.3: Conversation between the user and the bot

Image 4: The bot is asking for the feedback from the user about the entire conversation



thanks

feedback+did_that_help

utter_did_that_help

Did that help you? Could you provide a feedback ?

action_listen

/choose{"value":"positive"}

choose{"value":"positive"}

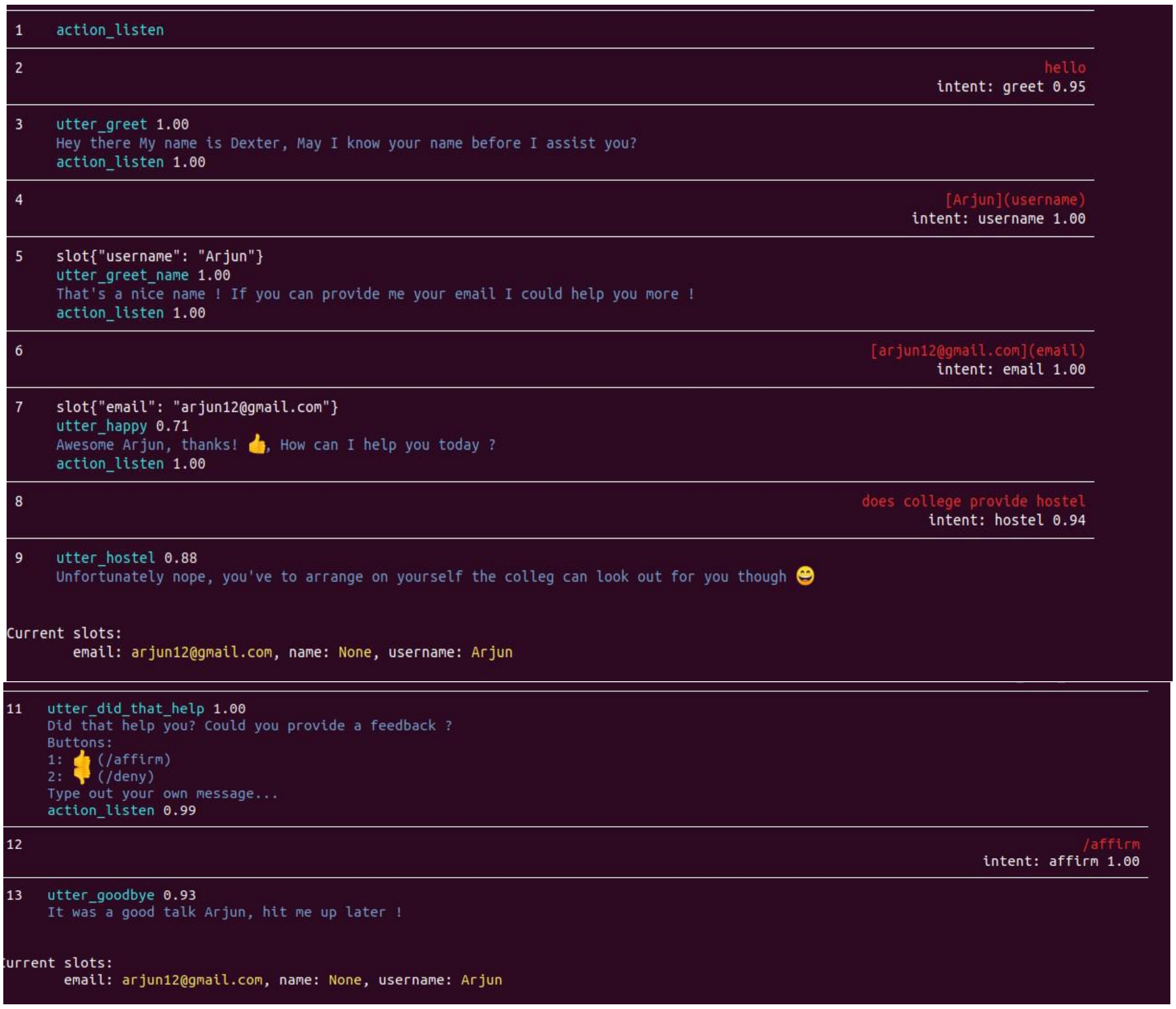
utter_goodbye

Bye ajinkya,see you later and thanks for stopping by !

action_listen

Figure 9.4: Conversation between the user and the bot

Interactive Training of Model in Rasa is given below:



```
1    action_listen
2                                                                                   hello
                                                                       intent: greet 0.95
3    utter_greet 1.00
     Hey there My name is Dexter, May I know your name before I assist you?
     action_listen 1.00
4                                                                      [Arjun](username)
                                                                   intent: username 1.00
5    slot{"username": "Arjun"}
     utter_greet_name 1.00
     That's a nice name ! If you can provide me your email I could help you more !
     action_listen 1.00
6                                                            [arjun12@gmail.com](email)
                                                                      intent: email 1.00
7    slot{"email": "arjun12@gmail.com"}
     utter_happy 0.71
     Awesome Arjun, thanks! 👍, How can I help you today ?
     action_listen 1.00
8                                                             does college provide hostel
                                                                      intent: hostel 0.94
9    utter_hostel 0.88
     Unfortunately nope, you've to arrange on yourself the colleg can look out for you though 😄

Current slots:
        email: arjun12@gmail.com, name: None, username: Arjun

11   utter_did_that_help 1.00
     Did that help you? Could you provide a feedback ?
     Buttons:
     1: 👍 (/affirm)
     2: 👎 (/deny)
     Type out your own message...
     action_listen 0.99
12                                                                                /affirm
                                                                      intent: affirm 1.00
13   utter_goodbye 0.93
     It was a good talk Arjun, hit me up later !

urrent slots:
        email: arjun12@gmail.com, name: None, username: Arjun
```

Fig 9.5: Interactive Model Training

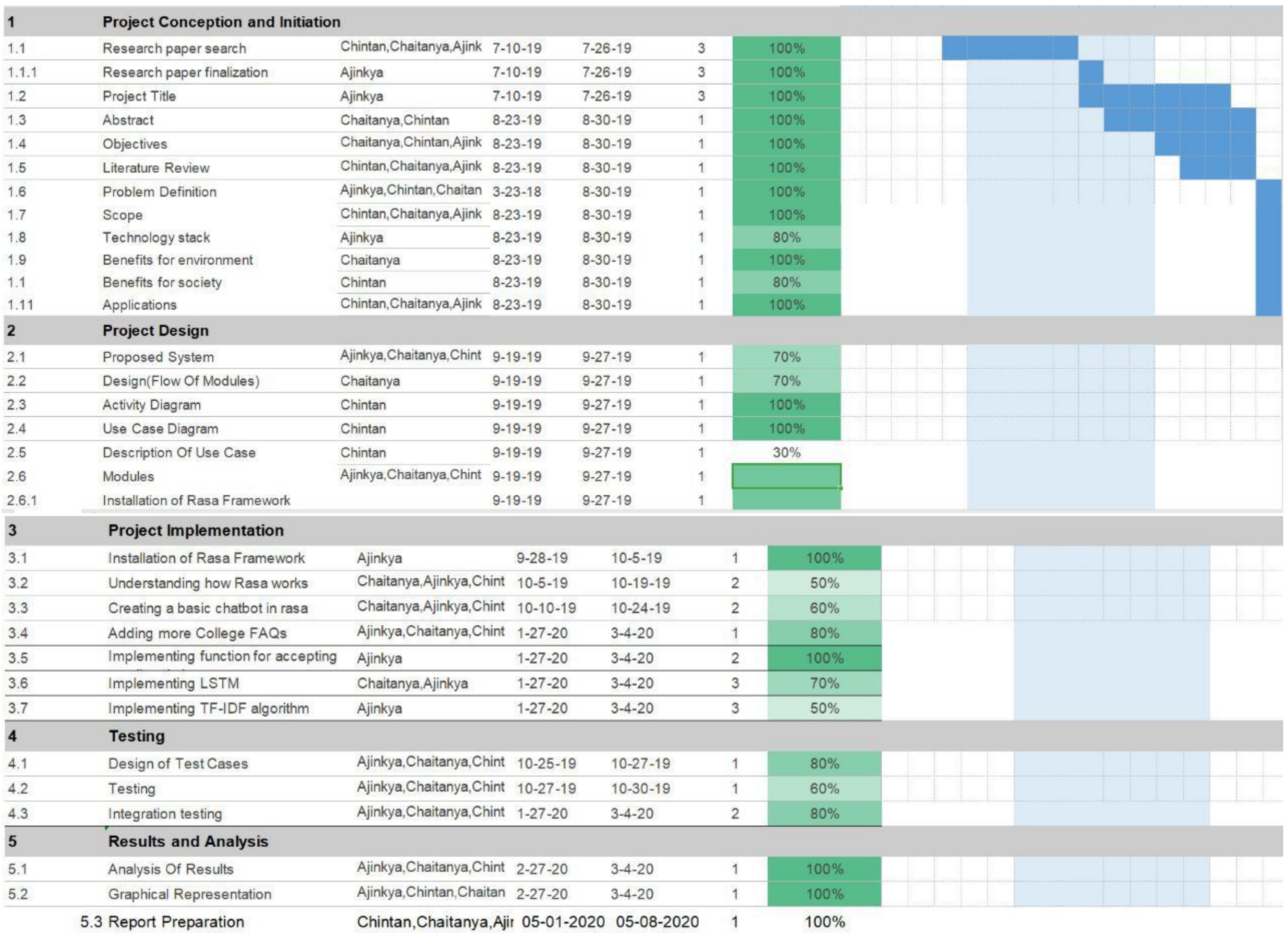
Chapter 10

Proposed Technology Stack

- Rasa Stack
- RNN (Recurrent Neural Networks)
- LSTM (Long Short Term Memory)
- Artificial Intelligence

Chapter 11

Project Timeline



1	**Project Conception and Initiation**					
1.1	Research paper search	Chintan,Chaitanya,Ajink	7-10-19	7-26-19	3	100%
1.1.1	Research paper finalization	Ajinkya	7-10-19	7-26-19	3	100%
1.2	Project Title	Ajinkya	7-10-19	7-26-19	3	100%
1.3	Abstract	Chaitanya,Chintan	8-23-19	8-30-19	1	100%
1.4	Objectives	Chaitanya,Chintan,Ajink	8-23-19	8-30-19	1	100%
1.5	Literature Review	Chintan,Chaitanya,Ajink	8-23-19	8-30-19	1	100%
1.6	Problem Definition	Ajinkya,Chintan,Chaitan	3-23-18	8-30-19	1	100%
1.7	Scope	Chintan,Chaitanya,Ajink	8-23-19	8-30-19	1	100%
1.8	Technology stack	Ajinkya	8-23-19	8-30-19	1	80%
1.9	Benefits for environment	Chaitanya	8-23-19	8-30-19	1	100%
1.1	Benefits for society	Chintan	8-23-19	8-30-19	1	80%
1.11	Applications	Chintan,Chaitanya,Ajink	8-23-19	8-30-19	1	100%
2	**Project Design**					
2.1	Proposed System	Ajinkya,Chaitanya,Chint	9-19-19	9-27-19	1	70%
2.2	Design(Flow Of Modules)	Chaitanya	9-19-19	9-27-19	1	70%
2.3	Activity Diagram	Chintan	9-19-19	9-27-19	1	100%
2.4	Use Case Diagram	Chintan	9-19-19	9-27-19	1	100%
2.5	Description Of Use Case	Chintan	9-19-19	9-27-19	1	30%
2.6	Modules	Ajinkya,Chaitanya,Chint	9-19-19	9-27-19	1	
2.6.1	Installation of Rasa Framework		9-19-19	9-27-19	1	
3	**Project Implementation**					
3.1	Installation of Rasa Framework	Ajinkya	9-28-19	10-5-19	1	100%
3.2	Understanding how Rasa works	Chaitanya,Ajinkya,Chint	10-5-19	10-19-19	2	50%
3.3	Creating a basic chatbot in rasa	Chaitanya,Ajinkya,Chint	10-10-19	10-24-19	2	60%
3.4	Adding more College FAQs	Ajinkya,Chaitanya,Chint	1-27-20	3-4-20	1	80%
3.5	Implementing function for accepting	Ajinkya	1-27-20	3-4-20	2	100%
3.6	Implementing LSTM	Chaitanya,Ajinkya	1-27-20	3-4-20	3	70%
3.7	Implementing TF-IDF algorithm	Ajinkya	1-27-20	3-4-20	3	50%
4	**Testing**					
4.1	Design of Test Cases	Ajinkya,Chaitanya,Chint	10-25-19	10-27-19	1	80%
4.2	Testing	Ajinkya,Chaitanya,Chint	10-27-19	10-30-19	1	60%
4.3	Integration testing	Ajinkya,Chaitanya,Chint	1-27-20	3-4-20	2	80%
5	**Results and Analysis**					
5.1	Analysis Of Results	Ajinkya,Chaitanya,Chint	2-27-20	3-4-20	1	100%
5.2	Graphical Representation	Ajinkya,Chintan,Chaitan	2-27-20	3-4-20	1	100%
5.3	Report Preparation	Chintan,Chaitanya,Ajir	05-01-2020	05-08-2020	1	100%

Fig 11.1: Gantt Chart

Chapter 12

Conclusions and Future Scope

The development of this chatbot will provide an easier way for students and any person to solve their queries faster and in an easier way. Not only it will help people to get their queries solved easily but it will also help reduce the work stress of the receptionist. The chatbot will be developed using RNN and LSTM, because of which the chatbot will be able to frame its answer if the answer for that particular question is not available in the database.

Integrate Voice Based Bot:

It can be converted into voice based bot.

Integrate Various Language in Bot:

We can include various languages so that we become easy for the user to communicate.

Deployed In Android and IOS:

It can be further deployed into an android / iOS application

Bibliography:

[1] Nudtaporn Rosruen and Taweesak Samanchuen "Chatbot Utilization for Medical Consul- tant System", The 2018 Technology Innovation Management and Engineering Science Inter- national Conference(TIMES-iCON2018)

[2]SathitPrasomphan, "Improvement of Chatbot in Trading System for SMEs by Using Deep Neural Network",2019IEEE4th International Conference on Cloud Computing and Big Data Analytics.

[3] Panitan Muangkammuen,Narong Intiruk, Kanda Runapongsa Saikaew, "Automated Thai-FAQ Chatbot usingRNN-LSTM".

[4] Rupesh Singh, Harshkumar Patel, Manmath Paste, Nitin Mishra, Nirmala Shinde, "Chat-bot using TensorFlow for small Businesses", Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant - Part Number: CFP18BAC-ART; ISBN:978- 1-5386-1974-2.

[5] Rohit Binu Mathew, Sandra Varghese, Sera Elsa Joy, Swanthana Susan Alex, "Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning", Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32- ART; ISBN: 978-1-5386-9439-8

Appendices

Appendix-A:Installation of Rasa Framework

Step-by-step Installation Guide

Install the Python development environment

Run the command on the terminal

```
$ sudo apt update

$ sudo apt install python3-dev python3-pip
```

Create a virtual environment (strongly recommended)

```
$ python3 -m venv --system-site-packages ./venv
```

Activate the virtual environment:

```
$ source ./venv/bin/activate
```

Install Rasa and Rasa X (Using given Command on terminal)

```
$ pip install rasa
```

After Successfully Installing Rasa, Download following Dependencies

```
$ git clone https://github.com/RasaHQ/rasa.git

$ cd rasa

$ pip install -r requirements.txt

$ pip install -e .
```

NLU Pipeline Dependencies

```
$ pip install -r alt_requirements/requirements_full.txt
```

SpaCy Dependencies

```
$ pip install rasa[spacy]

$ python -m spacy download en_core_web_md

$ python -m spacy link en_core_web_md en
```

Appendix-B: Creating a New Project and Training the model

Create a new project

A single command sets up a complete project for you with some example training data.

```
rasa init
```

Train a Model

The main command is:

```
rasa train
```

Interactive Learning

To start an interactive learning session with your assistant, run

```
rasa interactive
```

Talk to your Assistant

To start a chat session with your assistant on the command line, run:

```
rasa shell
```

To increase the logging level for debugging, run:

```
rasa shell --debug
```

Start a Server

To start a server running your Rasa model, run:

```
rasa run
```

Start an Action Server

To run your action server run

```
rasa run actions
```

Visualize your Stories

To open a browser tab with a graph showing your stories:

```
rasa visualize
```

Evaluate a Model on Test Data

To evaluate your model on test data, run:

```
rasa test
```

Create a Train-Test Split

To create a split of your NLU data, run:

```
rasa data split nlu
```

Start Rasa X

Rasa X is a toolset that helps you leverage conversations to improve your assistant.

You can start Rasa X locally by executing

```
rasa x
```

Acknowledgement

We have great pleasure in presenting the report on **“DEXTER” - The College FAQ Chatbot.** We take this opportunity to express our sincere thanks towards our guide **Dr Uttam D.Kolekar** & Co-Guide **Prof. Kaushiki Upadhyaya** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Mr. Kiran B. Deshpande** Head of Department,IT, APSIT for his encour- agement during progress meeting and providing guidelines to write this report.

We thank **Ms Kaushiki Upadhyaya** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1:Ajinkya Huddar
Student ID1:15104047

Student Name2:Chintan Suchak
Student ID2:16104069

Student Name3:Chaitaniya Bysani
Student ID3:16104045

Publication

Paper entitled **“Dexter- The College FAQ ChatBot”** is presented at **“IEEE”** by **“Ajinkya Huddar”** **“Chintan Suchak”** and **“Chaitanya Bysani**