UNIVERSITY CHATBOT

Minor project report-2020

A project report submitted in complete fulfillment of the requirements for Mini Project.

Integrated Post Graduate

by

Dara SravanKumar (2017IMT-030)
Suggula Jagadeesh (2017IMT-082)
Verra Dinesh (2017IMT-089)

under the supervision of-

Dr. Somesh Kumar

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015

2020

CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the report, **University ChatBot**, in complete fulfillment of the requirements for the **Mini Project** and submitted to the institution is an authentic record of our own work carried out during the period January 2020 to June 2020 under the supervision of **Dr. Somesh Kumar** . We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: Signatures of the Research Supervisors

ABSTRACT

University ChatBot is a virtual assistant (chatbot) which aims at making website navigation simple. It provides a human-like interaction rather than a computer generated response. Many people face problems while navigating through college websites in search of particular files or information,Different college websites have a different structure . They may differ on the basis of how they arrange their data/files, how they showcase information (fest updates, admission notifications), where they put up their fee links, etc..

Our chatbot helps in providing an easy solution for such problems,It answers user queries related to that particular website.A user (parents, students, faculty, visitors) can know information like the list of faculty members working in that university, the fee structure for different courses, different activities taking place, etc., just by asking it.

Our Chatbot can redirect a user to a webpage based on his needs.It can handle multiple users at a time providing efficient and accurate answers for each query,It is available 24/7.Any new/unanswered questions are stored in log files for further improvement.Overall, the chatbot reduces manual effort for users and saves their time .

This project is focusing on creating a chatbot to be used by students to get their queries responded easily from the college website. The University Chatbot has the capacity to make friendly conversations; respond the course and faculty details; give the link for the academic calendar; answer the frequently asked questions; calculate the fees based on the student's input; and give the timings, address, contacts, and events information of the departments like Administration, Library, Events, College fests and additional information.

ACKNOWLEDGEMENTS

We are highly indebted to **Dr. Somesh Kumar** and are obliged for giving us the autonomy of functioning and experimenting with ideas. We would like to take this opportunity to express our profound gratitude to them not only for their academic guidance but also for their personal interest in our project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within us. The nurturing and blossoming of the present work is mainly due to their valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. Our mentor always answered a myriad of our doubts with smiling graciousness and prodigious patience, never letting us feel that we are novices by always lending an ear to our views, appreciating and improving them and by giving us a free hand in our project. It's only because of their overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, we are grateful to our Institution and colleagues whose constant encouragement served to renew our spirit, refocus our attention and energy and helped us in carrying out this work.

(Danthala Deepak) (Dara Sravan Kumar) (Suggula Jagadeesh) (Verra Dinesh)

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AI	**Artificial Intelligence**
IT	**Information Technology**
NLP	**Natural Language Processing**
NLU	**Natural Language Understanding**
NLG	**Natural Language Generation**
AIML	**Artificial Intelligence Markup Language**
LSA	**Latent Semantic Analysis**
DIET	**Dual Intent Entity Transformer**
TED	**Transformer Embedding Dialogue**
HTTP	**HyperText Transfer Protocol**
SVC	**Support Vector Classifier**
API	**Application Programming Interface**
SVM	**Support Vector Machine**

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

This chapter includes the details of the background, our problem statement, the motivation and the objectives of our thesis. In this section we briefly describe our project - University Chatbot.

1.1 INTRODUCTION

Currently, the world is moving towards automation and chatbots are being used in every domain. Our vision is to create a chatbot specifically for college related queries with human-level interaction.
Our project - University ChatBot - is a virtual assistant (chatbot) which aims at making website navigation simple for college websites. It focuses mainly on college websites as people sometimes face problems in understanding the structure of the website and have difficulty in finding the desired files or information.

1.1.1 BACKGROUND

This project focuses on creating a chatbot to be used by students to get their queries responded easily from the college website. A chatbot is a program which can do real conversations with textual and/or auditory methods. Using Artificial Intelligence (AI), chatbots can simulate human conversations. There are two categories of chatbots. One category is command based chatbots where chatbots rely on a database of replies and heuristics. The user must be very specific while asking the questions so that the bot can answer. Hence, these bots can answer a limited set of questions and cannot perform function outside of the code. The other category is chatbots based on AI or machine learning algorithms, these bots can answer ambiguous questions which means the user does not have to be specific while asking questions. Thus, these bots create replies for the user's queries using RASA-framework which is built on mainly - Natural Language Processing (NLP).

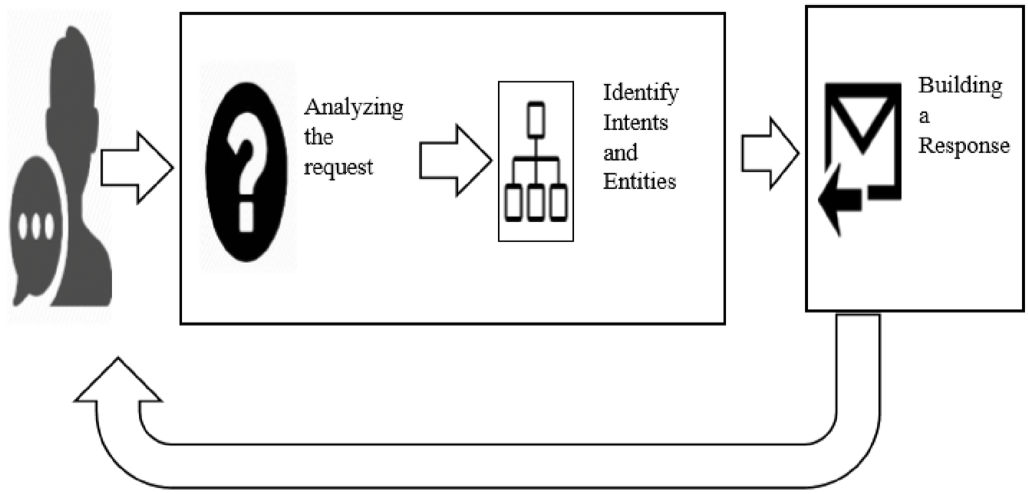

Figure 1: How a Chatbot Works [2]

1.1.2 MOTIVATION

As the Current day college/University websites are complicated and difficult to solve user queries, providing an AI based chatbot can make website navigation simple, easier and faster.
As students, we require many types of information regarding our college and university during our course. Sometimes getting this information is rather cumbersome and hectic.

We live in an age of Information Technology, where automation and simple procedures are easy to achieve. So, why have this long and unnecessary process to get this trivial information. We as Information Technology students are always looking forward to solving the problems around us using the technology that we learn and how to implement them to achieve ease of usage in real life. This is where we thought of using an intelligent chatbot delivering this information. No need to get into lengthy and hectic procedure

AI-powered chatbots are motivated by the need of traditional websites to provide a chat facility where a bot is required to be able to chat with users and solve queries. When a live agent can handle only two to three operations at a time, chatbots can operate without an upper limit which really scales up the operations.

Having a chatbot clearly improves the response rate compared to human support teams.Furthermore, a chatbot can automate the repetitive tasks. There can be some scenarios where a business or school receives the same queries in a day for many times and the support team must respond to each query repetitively. Lastly, the most important advantage of having a chatbot is that it is available 24/7.

Similarly, the University Chatbot is designed to help students to get their queries solved on a fingertip.

1.1.3 OBJECTIVES

The main objectives of this project are :

- To provide a user-friendly and fully functioning chatbot that makes navigation through a website simple.
- To ensure that the chatbot can answer queries and provide the user with information related to the institute.
- To ensure that the user can be redirected to the webpage of their choice without any hassle.
- To improve the current day university websites.
- To provide 24x7 availability
- To handle multiple users at a time providing efficient and accurate answers for each query.

1.2 RASA

Rasa is an open source machine learning framework for building AI assistants and chatbots. "**Rasa Action Server**" where you need to write code in Python, that is mainly used to trigger External actions like Calling Google API or REST API etc.
Rasa has two main modules:

1. **Rasa NLU** for understanding user messages
2. **Rasa Core** for holding conversations and deciding what to do next

Rasa NLU — This is the place, where rasa tries to understand User messages to detect **Intent** and **Entity** in your message. Rasa NLU has different components for recognizing intents and entities, most of which have some additional dependencies.

1. Spacy (You need to install it separately)
2. Tensorflow (By Default available with Rasa)

Rasa Core — This is the place, where Rasa tries to help you with contextual message flow. Based on User message, it can predict dialogue as a reply and can trigger Rasa Action Server.Rasa internally uses Tensorflow

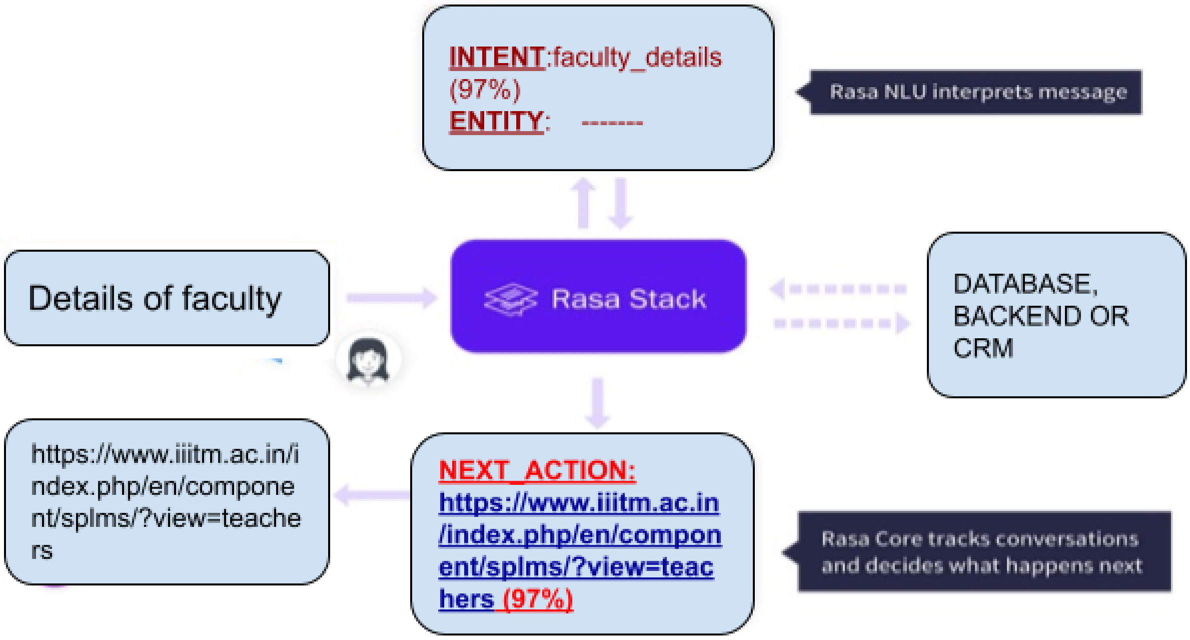

Figure 1.2 Rasa Stack

1.3 PYTHON

We have used PYTHON to implement our project. PYTHON is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991,It was released in the year 1991 and designed by Guido van Rossum. Python considers whitespace significant and it emphasizes code readability by design.

We have used Rasa-Framework for the development of the model used in University chatbot. Rasa is an open source machine learning framework for building AI assistants and chatbots which is primarily built on python and we use python in Rasa-Action-Server.
Key Features - High community support for supporting and helping others with their projects. Built-in framework for web development. Version - PYTHON (3.6.4)

1.4 LITERATURE REVIEW

- Eliza is considered as the first Chatbot, which works on the pattern matching system. It was developed by Joseph Weizenbaum in 1964. ALICE is a rule-based chatbot based on the Artificial Intelligence Markup Language (AIML). It has more than 40,000 categories, where each category has a combination of pattern and its response.
- Md.Shahriar Satu and Shamim-Al-Mamun showed the review of applications of the Chatbot which are developed using the AIML scripts. They said that AIML based chatbots are easy to implement, they are lightweight and efficient to work. Their paper gives the detailed information about the different applications of the chatbots.
- Thomas N. T. and Amrita Vishwa designed an AIML and LSA (Latent Semantic Analysis) based chatbot to provide the customer care service over the E- commerce websites. Their approach shows we can improve the chatbot ability by adding other models to it. In the android operating system, we can implement the chatbot using the various approaches. One of the approaches is shown by Rushabh Jain and Burhanuddin Lokhandwala in their Android based Chat-Bot paper.
- Emanuela Haller and Traian Rebedea, "Designing a Chat-bot that Simulates and Historical Figure", IEEE Conference Publications, July 2013. There are many applications that are incorporating a human appearance and intending to simulate human dialog, but in most of the cases, the knowledge of the conversational bot is stored in a database created by human experts. However, very few researches have investigated the idea of creating a chat-bot with an artificial character and personality starting from web pages or plain text about a certain person.
- Maja Pantic, Reinier Zwitserloot, and Robbert Jan Grootjans, "Teaching Introductory Artificial Intelligence using A simple Agent Framework", IEEE Transactions on Education, Vol. 48, No. 3, August 2005. This paper describes a flexible method of teaching introductory artificial intelligence (AI) using a novel, Java-implemented, simple agent framework developed specifically for the purposes of this course.
- MA Ulrich Gnewuch Prof. Alexander Mädche "Early chatbots were not more than simple command-line interfaces. Over the years, they have been endowed with all kinds of visual appearances. However, it is unclear if chatbots really need rich visual representations.

CHAPTER 2

DESIGN DETAILS AND IMPLEMENTATION

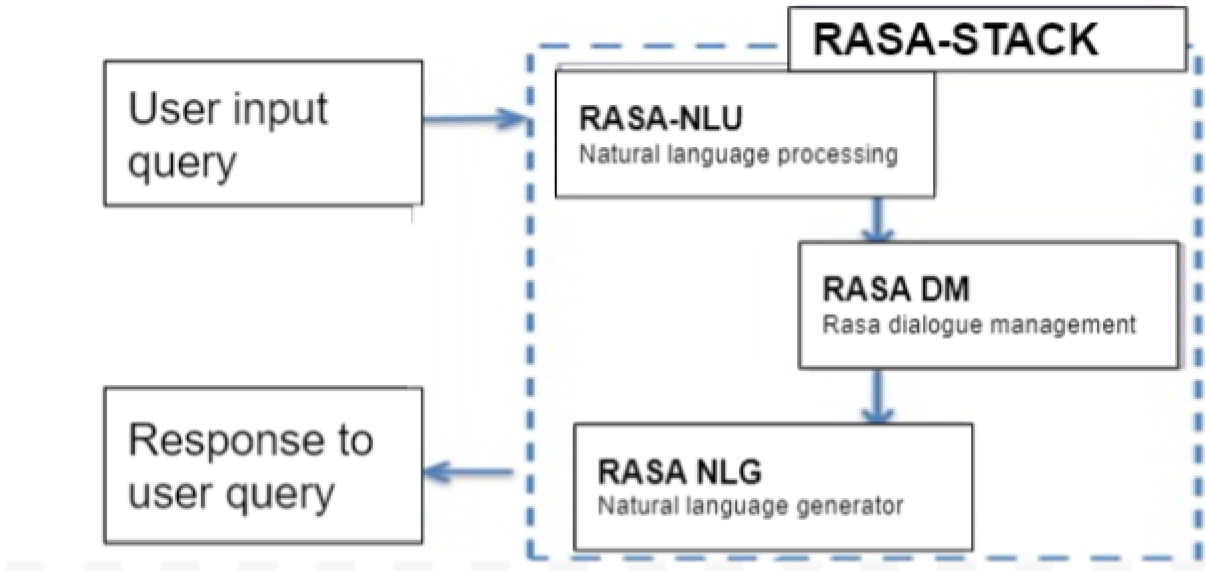

Figure 2 Work Flow of RASA Stack

2.1 NATURAL LANGUAGE UNDERSTANDING (RASA-NLU)

Rasa NLU is an open-source natural language processing tool for intent classification, response retrieval and entity extraction in chatbots.It comprises loosely coupled modules combining a number of natural language processing and machine learning libraries. There are some predefined pipelines like spacy_sklearn, tensorflow_embedding, Supervised Embeddings, with sensible defaults which work well for most use cases.

Rasa NLU uses Support Vector Classifier (SVC) algorithm for Intents recognition and Entity extraction. The classifier is applied to vector representations of words, where each word is represented by a dense numeric vector. Rasa provides pre-trained models using publicly available training datasets; Chatbot developers can also update the model with additional examples.

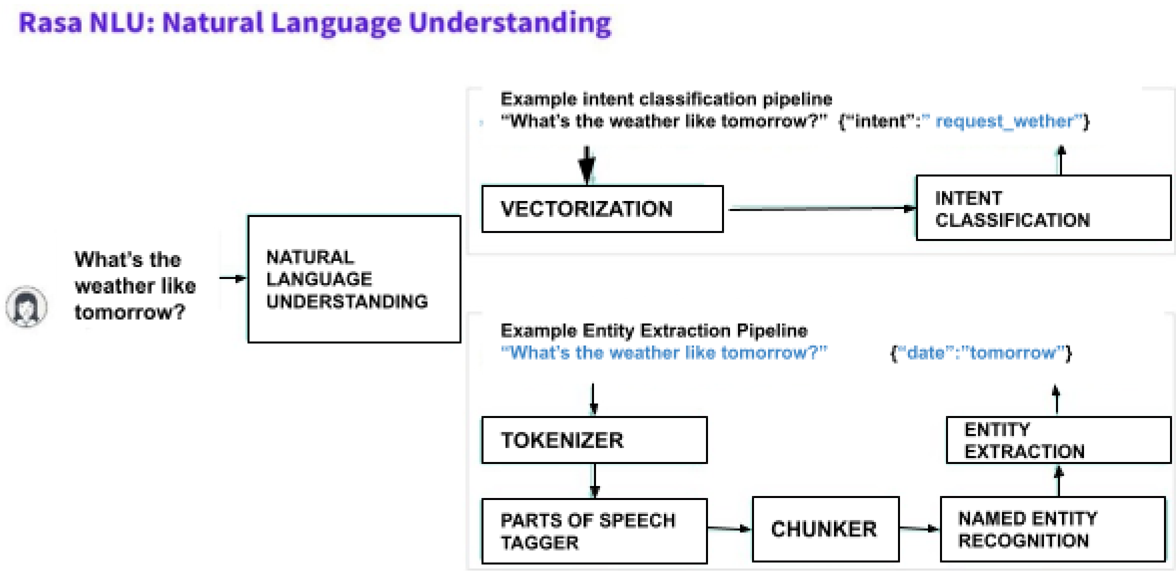

Figure 2.1 Natural Language Understanding

2.1.1 Pipelines

Incoming messages are processed by a sequence of components. These components are executed one after another in a so-called processing pipeline defined in your config.yml. Choosing an NLU pipeline allows you to customize your model and finetune it on your dataset.

We had created our own NLU pipeline that allows us to customize our model and finetune it on our dataset.

Tokenizers split text into tokens. If you want to split intents into multiple labels, e.g. for predicting multiple intents or for modeling hierarchical intent structure

First, the text is tokenized and parts of speech (POS) annotated using the Whitespace Tokenizer which uses whitespaces as a separator.
Then the Regex Featurizer looks to create a vector representation of the user message using regular expressions. Then the LexicalSyntacticFeaturizer creates lexical and syntactic features for a user message to support entity extraction.

Next, the CountVectorsFeaturizer, creates a bag-of-words representation of generated tokens and features. Next, DIETClassifier the Dual Intent Entity Transformer (DIET) used for intent classification and entity extraction

EntitySynonymMapper maps synonymous entity values to the same value.

Finally the Response Selector predicts a bot response from a set of candidate responses.

```
# Configuration for Rasa NLU.
# https://rasa.com/docs/rasa/nlu/components/
language: en
pipeline:
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: "char_wb"
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 100
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
```

Figure 2.1.1 code snippet of pipelines used

2.2 NATURAL LANGUAGE GENERATION (NLG)

Natural Language Generation (NLG) is a subdivision of Artificial Intelligence (AI) that aims to reduce communicative gaps between machines and humans. The technology typically accepts input in non-linguistic format and turns it into human understandable formats like reports, documents, text messages etc.

Most of the weather forecasting systems use Natural Language Processing to interpret the numerical values that are received as an input from supercomputers. For converting this data into a language (text, audio, print, or any other form) that humans understand, Natural Language Generation can be used. The system will continue to receive the values and NLG interprets them into human understandable format.

2.3 DIALOGUE HANDLING

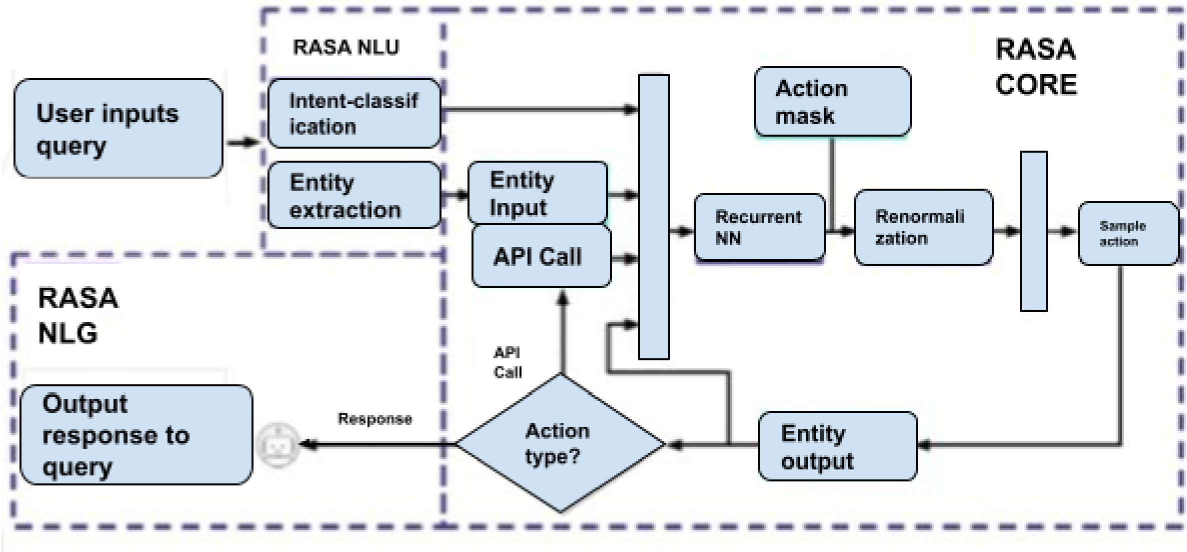

Figure 2.3 Dialogue Handling

Rasa Core (dialogue handling) predicts which action to take from a predefined list. An action can be a simple utterance, i.e. sending a message to the user, or it can be an arbitrary function to execute. When an action is executed, it is passed a tracker instance, and so can make use of any relevant information collected over the history of the dialogue: slots, previous utterances, and the results of previous actions. Actions cannot directly mutate the tracker, but when executed may return a list of events. The tracker consumes these events to update its state. There are a number of different event types, such as SlotSet, AllSlotsReset, Restarted, etc.

2.3.1 Policies

We used a set of predefined policies in rasa core to train the chatbot.

Memoization Policy
The MemoizationPolicy just memorizes the conversations in your training data. It predicts the next action with confidence 1.0 if this exact conversation exists in the training data, otherwise it predicts None with confidence 0.0.

TED Policy
The Transformer Embedding Dialogue (TED) Policy is a pre-defined architecture, which comprises the following steps:

- concatenate user input (user intent and entities), previous system actions, slots and active forms for each time step into an input vector to pre-transformer embedding layer ,feed it to the transformer.
- apply a dense layer to the output of the transformer to get embeddings of a dialogue for each time step
- apply a dense layer to create embeddings for system actions for each time step.

Mapping Policy

The MappingPolicy can be used to directly map intents to actions. The mappings are assigned by giving an intent the property

The epochs represent the no of times the chatbots needed to be trained with the data , It is set to be 200.

```
# Configuration for Rasa Core.
# https://rasa.com/docs/rasa/core/policies/
policies:
  - name: MemoizationPolicy
  - name: TEDPolicy
    max_history: 5
    epochs: 200
  - name: MappingPolicy
```

Figure 2.3.1 code snippet of policies used

A chatbot possessing NLG ability would mean that the chatbot knows what exact and clear response (message) to generate for a corresponding user message.However, if the user had already provided this information in the message then the chatbot should not ask for the same information again.'
Here we can see that the response must be coherent, meaningful, contextual, complete, non-repetitive and clear. This mandates that the response cannot be static but must be dynamic.

Basically, there are two ways that any chatbot can generate a response. One of the ways this has been solved is by using a Dialog management system. Another way is for different input messages the predictive model can be trained to decide what to say next, contextually. Rasa Core is one such Dialog Management system for NLG.
Recent research in NLG has uncovered Deep Learning algorithms, especially Neural Dialog Generation Models such as Sequence to Sequence, that are more effective in predicting responses to natural language conversations. This would tremendously improve the comprehension abilities of machines and would also enable chatbots to handle question answering problems.

RASA also offers a built-in templated based NLG. However, it also allows you to connect to an external HTTP server for NLG. What happens in that server is up to you, and it could be a neural network based NLG server.

2.4 CHATBOT ARCHITECTURE:

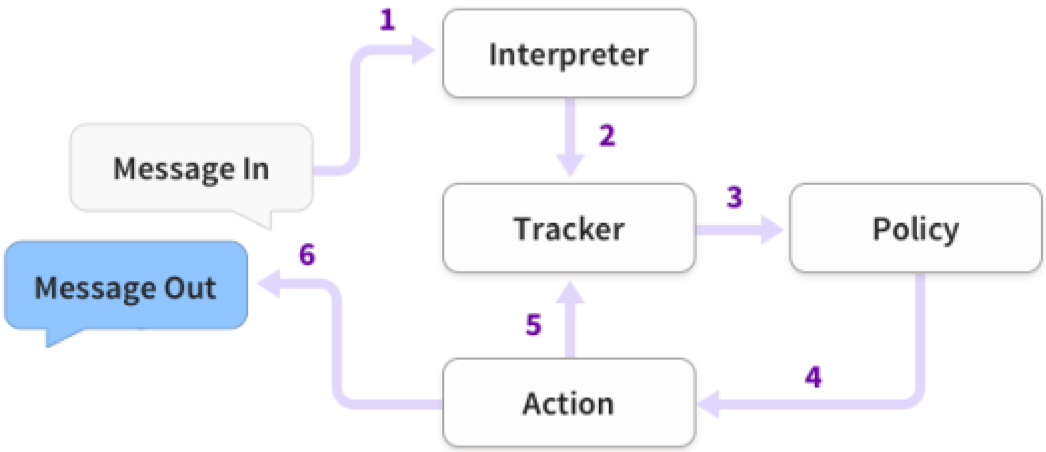

Figure 2.4 Architecture of Chatbot

The message is received and passed to an INTERPRETER which converts it into a dictionary including the original text, the intent, and any entities that were found. This part is handled by NLU.

The TRACKER is the object which keeps track of conversation state. It receives the info that a new message has come in.

The policy receives the current state of the tracker.The policy chooses which action to take next.The chosen action is logged by the tracker.A response is sent to the user.

2.5 INTENT CLASSIFICATION

We have a new pipeline which is totally different from the standard Rasa NLU approach. It uses very little memory, handles hierarchical intents, messages containing multiple intents, and has fewer out-of-vocabulary issues. And in principle it can do intent recognition in any language.

The standard way we've been doing intent classification since Rasa NLU was released is to represent sentences as a sum of word vectors, and then train a classifier on that representation. We run regular benchmarks on a dozen different datasets, where we try different word vectors and classifiers to see what really moves the needle. Mostly it's the quality (or appropriateness) of your word vectors that matters, and using a neural net instead of a support vector machine (SVM) doesn't make any difference.

Our new TensorFlow embedding pipeline does almost the exact opposite. It doesn't use pre-trained word vectors, and should work on any language.
Our new embedding pipeline doesn't use pre-trained vectors, but instead learns embedding for both the intents and the words simultaneously. And instead of training a classifier, these embeddings are used to rank the similarity between an input sentence and all of the intents. This means you aren't stuck with out-of-the-box pre-trained word vectors, but learn your own specifically for your domain.

```
session_config:
  session_expiration_time: 5
  carry_over_slots_to_new_session: false
intents:
- greet
- college_query
- faculty_query
- course_query
- out_of_scope
- goodbye
- affirm
- deny
- mood_great
- mood_unhappy
- bot_challenge
```

Figure 2.5 code snippet of Intents created

2.6 ROLE OF NLU AND CORE

NLU is a sub-field of NLP which handles a narrow but complex challenge of converting unstructured inputs into a structured form which a machine can understand and act upon.

In this section, I would like to explain Rasa in detail and some terms used in NLP which you should be familiar with.

Intent: This tells us what the user would like to do.

Ex : Raise a complaint, request for refund etc

Entities: These are the attributes which give details about the user's task. Example: Complaint regarding service disruptions, refund cost etc.

Confidence Score : This is a distance metric which indicates how closely the NLU could classify the result into the list of intents.

Rasa NLU's job is to accept a sentence/statement and give us the intent, entities and a confidence score which could be used by our bot. Rasa basically provides a high level API over various NLP and ML libraries which does intent classification and entity extraction. These NLP and ML libraries are called a backend in Rasa which brings the intelligence in Rasa. These are some of the backends used with Rasa.

"Text": " What is the location of college? "

-intent: college_details

-entities:college

Confidence score : 0.84 #(this could vary based on our training set)

Figure 2.6 example of user query

CHAPTER 3

RESULTS AND DISCUSSION

3.1 EARLY TESTING AND FINDINGS

In the beginning of our project we wanted to test the first version of our chatbot on our friends . This was late in the fall and most of our friends were familiar with a lot of the answers our chatbot could provide.

We wanted to test this early version of the chatbot to get input on what the chatbot could and could not answer in the future. After the test was completed we had a short interview with the participants.

The main purpose for this test was to see how the participants interacted with the initial version of the chatbot and find out if the chatbot could be suitable to find the information they needed. Before the testing we also carried out a pilot test to find immediate flaws in the plan.

3.1.1 RESULTS FROM EARLY TESTING

The first participant enjoyed talking to the bot, but stressed the fact that you had to "talk like "a dummy" for it to understand what you were asking. The participant pointed out that this really would have come in handy in his first weeks at the university, as he didn't always know who to ask - especially if he was in a hurry. He pointed out that the chatbot needs to get more features like tell you details of faculty members etc.

The second participant was a bit frustrated that the chatbot wasn't flexible enough . "I don't like having to guess what questions to ask". He would like more instructions to know how to get more out of the chatbot.

The third participant also had problems with understanding what the chatbot could do. When given a hint for what the chatbot could do, the chatbot did not function properly.

Here we tried to restart the system and then the chatbot displayed it´s welcome message about what it could do. Afterwards it was more clear what the participant could ask for.

3.1.2 RE-DESIGN OF THE PROTOTYPE

These findings gave us a lot of insight in where the chatbot needed to be changed. E.g. adding a proper welcome message, defining the chatbots' limitations and presenting this to the user.
It's important to define goals and expectations so that your chatbot has a clear purpose. Knowing the capabilities and limitations of the system, before it crashes. The test showed that it was hard to ask the 'right' questions, we
therefore added more 'AI ques' to simplify the interaction. We also used the principles for designing conversational agents.

3.2 EVALUATING THE CHATBOT

Due to time and capacity during this project we decided on including five participants acting as evaluators. The number of participants is also chosen on the basis that five participants can contribute to finding 80% of the usability flaws. The evaluation was formed as a formative usability test where the goal is to look at metrics that are more qualitative than quantitative.

In the evaluation we wanted to combine small semi-structured interviews with the users executing tasks because this could give us more information about the experience beyond the metrics.

3.2.1 THE EVALUATION PLAN

SET UP	**Candidates** : Five randomly picked evaluators, the only criteria is that they have to be students from IFI. **Context** : In the Institute for informatics building
WARMING UP	Have you talked with a chatbot before? If <u>Yes</u> : What type of chatbot? How do you feel about getting information from a chatbot? Do you consider the information as more or less reliable?

TASKS	**Scenario** : Imagine you are a new user who wants to know about IIITM. Use the chatbot and try to figure out where the location of college is, No. of hostels in the college. Later you feel like knowing about the faculty of IIITM. **Tasks** : Use the chatbot to find out: *About the faculty of IIITM?* Have a chat with the chatbot

3.2.2 FINDINGS FROM THE EVALUATION

All of our participants reported that they had interacted with chatbots before, but had very little knowledge about how they worked. They found the chatbot to be nice to interact with and enjoyed that he had a friendly conversation. One of the participants said that he did not want a chatbot that felt too 'human-like', and that the prototype did not feel 'human-like at all. This became clear when the same error message appears several times during the test.

They found it hard to get the right answer but when they did they were very satisfied with the answers. "It was a good answer when I finally got the right one..".

It was pointed out that the chatbot was not a smart chatbot, but that it provided the most necessary information sparing them from precious time spent on 'Google'.

They also reported that they trusted the answers they got, and they all pointed out that it was good that the chatbot provided a source along with the information it gave.

One of the participants also stated: "I liked that the chatbot was casual" . I don't want a formal and boring chatbot, so I could have tried to find it on the university's web-pages." It was also pointed out that it was preferably that the chatbot could provide diverse information, "Usually, the information is so spread that you don't know where to look"

3.3 TESTING USING RASA-X

Rasa X is a tool that helps you build, improve, and deploy AI Assistants that are powered by the Rasa framework. Rasa X runs on your own computer and you can deploy it to your own server. None of your conversations or training data are ever sent anywhere.

Some things, like inspecting and annotating conversations, are much easier with a UI. Rasa X focuses on those use cases and not on replacing things that are easier to do in code.

Rasa X is a bot framework emulator is a desktop application that can be used to test the bots created with RASA Framework. This tool allows us to test our bots locally or even remotely. It emulates a normal conversation but also shows information that is sent and received with each message. The emulator also gives us the opportunity to test our chatbots locally before they are pushed online and in reach of users.

We can test the accuracy of the bots replies when asking different questions to see what responses may need to be updated/modified. Having a sandbox to really dig into the bot and see how it performs with a range of different queries has helped greatly in the development of this project. This information gives an insight into how the bot is working and has been vital for the testing of our project.

Here is a screenshot of a conversation between our chatbot and a tester. Rasa-X provides a user interface to view the different conversations our chatbot is having with different users all at one place.

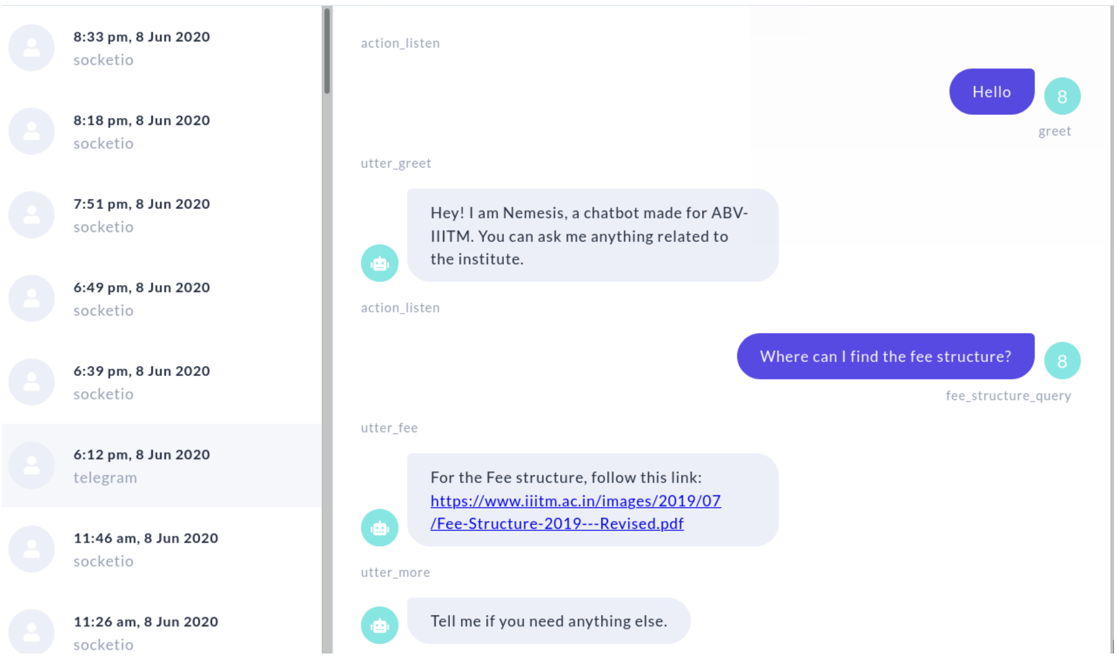

Figure 3.3.1 stored conversations in RASA-X

Rasa-X also provides us with a facility to add more training data gathered from real-world conversations. The picture below shows different intents as classified by our chatbot along with their confidence scores. We can manually add the data to our training set or delete it if it was classified incorrectly.

Sentence	Predicted Intent and Confidence
HI	greet (0.98)
Okay bye	goodbye (0.86)
Where is the college located?	location_query (0.99)
Where can I download the academic calendar	acal_query (0.95)

Figure 3.3.2 confidence score of intent classification

We can also train new models with the updated training data directly from the Rasa-X server. Below is a screenshot of the different models that were trained during the testing of our chatbot.

Model name	Created on	State
20200608-131511	Mon Jun 08 2020 18:45:11 GMT+0530	Active
20200608-131451	Mon Jun 08 2020 18:44:51 GMT+0530	
20200607-084317	Sun Jun 07 2020 14:13:17 GMT+0530	
20200607-084254	Sun Jun 07 2020 14:12:54 GMT+0530	

Figure 3.3.3 trained models

3.4 DISCUSSION

When testing the chatbot after all changes, we got findings suggesting that the participants did not have a problem with getting information from a chatbot instead of a human. The information that they got was not seen as less trustworthy, this could be supported by the fact that the chatbot provided a source for the information it gave.

It has been interesting to investigate how the participants interacted with the chatbot and how they reported on it afterwards. Our findings have some indicators leading towards that a chatbot could be a good alternative for acting as a helpful friend for freshmans at a new school.

Still we have to stress the fact that the chatbot was not very intelligent and that the evaluators had to adjust their language to match the chatbots.

Because of the scope of the project we did not have time to conduct as much user testing and re-design to the chatbot as we would have liked. This has an impact on the validity of our research. Through the project we have touched on some theory when making the chatbot, but this should also have a larger focus for higher validity.

Even though the participants trusted the information given in this project we cannot say that people trust a chatbot as much as they trust a human being. There are also biases in our project, one of them is that all the students that we included in the project already knew a lot of the answers the prototype could provide.

Another bias is that the information the chatbot provides could be seen as “casual” and are not crucial and/or vital This could have had an impact on the results regarding trustworthiness. With that being said we also think that some of our findings could give some insights into how a very small group of people think about using a chatbot to gain information about the college.

CHAPTER 4

CONCLUSION

4.1 CONCLUSION

To summarize, University ChatBot - is a virtual assistant (chatbot) which aims to make website navigation simple for college websites.
University Chatbot is an Artificial Intelligence based Chatbot which mainly uses Natural Language Processing(Natural Language Understanding ,Natural Language Generation) and Machine Learning Techniques,which is built on RASA.

University Chatbot is helpful in guiding users(students/faculty/guests/parents) with correct and most up to date sources of information. Users can get the information at their fingertips rather than visiting a college . It improves efficiency by taking over tasks for which humans are not essential and it is available 24x7 all the time.

University chatbot improves the response rate compared to human support teams.Furthermore, It can automate the repetitive tasks and can handle multiple users at a time.
Through this chatbot,college websites become really handy and can access information faster, reliable from anywhere and anytime.

4.2 SCOPE FOR FUTURE WORK

To improve the current functionalities of University Chatbot, in the future,
The scope of the chatbot can be increased by inserting data for all the departments, training the bot with varied data, testing it on with more users, and based on that feedback inserting more training data to the bot.

Some of the new features which can be added to the bot are

- Speech recognition feature through which students can ask their queries verbally and get the answers from the bot.
- Integration with multiple channels such as various social media platforms like Skype,Whatsapp, Facebook and Twitter.
- Handling context aware and interactive queries in which bot will be aware of the context of an ongoing conversation with a student.
- Making API’s for storing attendance,library management ,fees payments, and then integrating them with the chatbot.

REFERENCES

- Rasa Basics, https://rasa.com/docs/rasa/user-guide/rasa-tutorial/
- The Rasa Core Dialogue Engine, https://rasa.com/docs/rasa/core/about/
- Rasa NLU: Language Understanding for Chatbots and AI assistants, https://rasa.com/docs/rasa/nlu/about/
- Build and Deploy a Chatbot in Minutes using Rasa https://www.analyticsvidhya.com/blog/2019/04/learn-build-chatbot-rasa-nlp-ipl/
- A. Storman, "5 Chatbot Challenges and How to Overcome Them", Chatbots Magazine, 2016. https://chatbotsmagazine.com/5-chatbotchallenges-and-how-to-overcome-them-caccc3a26d7c.
- Deploying Your Rasa Assistant,https://rasa.com/docs/rasa/user-guide/how-to-deploy/http://dspace.calstate.edu/bitstream/handle/10211.3/210198/College%20Enquiry%20Chatbot%20Report_v12.pdf?sequence=1
- RASA master class https://www.youtube.com/watch?v=rlAQWbhwqLA&list=PL75e0qA87dlHQny7z43NduZHPo6qd-cRc