AI CHATBOT FOR UNIVERSITY ADMISSION

Group Members:

MUHAMMAD MOBEEN TAHIR	**191210**
AREEJ GUL	**191213**

BACHELOR OF COMPUTER ENGINEERING

FALL 2019 - 2023

PROJECT SUPERVISOR

USMAN HAMEED

ASSISTANT PROFESSOR

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

A Final Year Project-III Report

Presented to

AIR University

in partial fulfillment of the requirements

for the Degree of

Bachelor of Computer Engineering

AI CHATBOT FOR UNIVERSITY ADMISSION

SUBMITTED BY

MUHAMMAD MOBEEN TAHIR 191210

AREEJ GUL 191213

SPRING-2023

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

June 2023

Approval for Submission

It is to certify that the project report titled
" AI CHATBOT FOR UNIVERSITY ADMISSION "
has met the required standard for submission
in partial fulfillment of the requirements
for the award of degree of
Bachelor of Computer Engineering
at Air University, Islamabad.

Project Supervisor

_____

USMAN HAMEED
Assistant Professor

Head of Department

_____

Dr. Ashiq Hussain

ABSTARCT

This report outlines the development of a chatbot for university admission using Rasa natural language processing (NLP) and machine learning algorithms. With the increasing demand for 24/7 customer support, organizations have been turning to chatbots as a cost-effective and efficient solution. Our proposed study aims to replace traditional customer support models with an intelligent chatbot that can serve as a virtual helpdesk for university-related queries. To build the chatbot, we collected data on user demands and interactions and used machine learning algorithms to train the model. We experimented with several popular algorithms, including Decision Trees, Support Vector Ma- chines, and Random Forest, to find the best-performing model. Our results show that Random Forest outperforms the other algorithms for university admission purposes, responding well to both normal and even spun text. The chatbot can assist users with university-related queries, including ad- mission schedules, fee structures, offered programs, and other relevant informa- tion. The model can guide users through the university website, provide relevant links, and reduce the traffic to physical admission offices. Using Rasa NLP and dialog flow, the chatbot can understand natural lan- guage and engage in a conversation with users. The NLP toolkit enhances the chatbots ability to interpret user queries and provide relevant responses. The proposed chatbot is a cost-effective and efficient solution for universities, as it can reduce the workload on support staff and improve communication with stu- dents. The proposed chatbot for university admissions purposes has the potential to revolutionize the way universities interact with students. By leveraging the power of Rasa and machine learning, we have developed an intelligent chatbot that can assist students queries and provide a seamless user experience.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 The Importance of Project

In todays world, customer support services are in high demand, and organi- zations must ensure that their support centers are available 24/7 to cater to the needs and demands of their customers. However, this increases the cost of human resources, as people have to work round the clock to provide uninterrupted support services. In an effort to reduce costs and time, many organizations are seeking to automate their support processes. The proposed study aims to provide a solution to this problem by developing a chatbot that can handle support and help requests in universities. The use of a chatbot can reduce the workload of human resources and provide a more efficient and cost-effective way of delivering support services to students and visitors to the university. The chatbot can be used to provide assistance and guidance related to university websites, admission procedures, and other university-related matters. The data collected from user interactions can be used to train the chatbot using machine learning algorithms, such as Decision Trees, Support Vector Machines [1], and Random Forests. Based on the results of our study, the Random Forest algorithm was found to be the most effective in handling university-related support queries. It outper- formed both the Decision Tree and Support Vector Machine algorithms in terms of response time and accuracy. The chatbot can be used to guide visitors to the university website and provide links to relevant information, reducing the need for physical visits to the admission office.

By implementing the proposed chatbot, universities can significantly reduce the

workload of their support centers and improve their communications with students and visitors. The chatbot can provide support and guidance 24/7, which can increase the satisfaction levels of students and visitors. Moreover, the use of machine learning algorithms can enable the chatbot to learn from previous interactions, thereby improving its performance over time. Overall, the proposed study provides a viable solution to the problem of increas- ing costs and time associated with human resources in support centers. The use of a chatbot can significantly reduce these costs and provide an efficient and cost-effective way of delivering support services in universities. By implement- ing the proposed chatbot, universities can improve their communications with students and visitors, and reduce the need for physical visits to the admission office.

1.2 Chatbot

1.2.1 What is Chatbot

A chatbot is an artificial intelligence (AI) system designed to mimic human conversation and interact with users via text or voice [2]. Chatbots use natural language processing (NLP) to understand and respond to user inputs. The communication between a chatbot and a user is usually in a back-and-forth manner, similar to a game of tennis. Chatbots can be built using different techniques, such as AIML, retrieval-based models, and generative models. AIML is a popular and relatively straightfor- ward approach to building chatbots. Retrieval-based models use a database of predefined responses and an algorithm to choose the most suitable response based on the input and context. Generative models, on the other hand, use ma- chine learning techniques to generate responses based on the input and context. [3] The history of chatbots dates back to 1964 when the first chatbot, ELIZA, was created. Over the years, chatbots have evolved to become more sophisticated and capable of handling a wide range of user cases. Nowadays, chatbots are commonly used in customer service, information acquisition, and other related purposes. The classification of chatbots can be based on three dimensions. The first di- mension is "building approaches," which distinguishes between retrieval-based models and generative models. Retrieval-based models are easier to build and do not make grammatical mistakes, but they are unable to handle unseen cases where no predefined response exists [4]. Generative models, on the other hand, are capable of generating responses based on machine learning techniques. The second dimension is related to the level

of intelligence of chatbots. Simple chatbots use predefined responses to user inputs, while more advanced chatbots use machine learning techniques to improve their responses based on previous interactions with users. The third dimension is related to the level of autonomy of chatbots, which can be fully automated or supervised by a human operator. Chatbots have many applications in various industries, such as healthcare, fi- nance, and education. In healthcare, chatbots can be used for patient engage- ment and education, while in finance, chatbots can be used for financial advice and support. In education, chatbots can be used as intelligent tutors to assist students with their studies. In conclusion, chatbots are an important AI technology that can improve com- munication and engagement with users. As technology continues to advance, chatbots will become even more sophisticated and capable of handling complex user cases.

1.2.2 Types of Chatbot

There are two Main types of Chatbot:

- Rule Based Chatbot.
- AI Based Chatbot.

1.2.2.1 Rule Based Chatbot

Rule-based chatbots, also known as decision tree bots, employ a set of predefined rules to determine the types of problems they are familiar with and can provide solutions for. These bots use a flowchart-like structure to map out conversations, anticipating user queries and providing appropriate responses based on the defined rules. Rule-based chatbots can use simple or complex rules but are limited to answer- ing questions within their pre-defined ruleset. They do not learn through inter- actions, and their performance is limited to the situations they have been trained for. While rule-based bots may have a less flexible conversational flow, they provide the advantage of delivering a predictable experience, unlike AI-based chatbots. One of the major advantages of a rule-based chatbot is its faster training time, making it less expensive than other chatbot alternatives. They are also efficient at detecting and removing weeds, meaning irrelevant or out-of-scope inquiries are quickly identified and dealt with. Moreover, they integrate seamlessly with legacy systems, making them ideal for organizations that already have estab- lished systems in place. Rule-based chatbots are highly accountable and secure, providing an added layer of protection

for sensitive information. They can also incorporate interactive elements and media to enhance the user experience. It is important to note that while rule-based chatbots offer several benefits, their effectiveness is limited to the scope of the defined rules. As such, they require constant maintenance and updating to ensure they remain relevant and effective in addressing user queries. Additionally, they may not be suitable for handling complex queries that require a more dynamic approach. In conclusion, rule-based chatbots provide an efficient and cost-effective means of addressing routine user queries. Their predictability and accountability make them ideal for organizations seeking a secure and reliable conversational inter- face. However, they require ongoing maintenance to remain effective and may not be suitable for more complex inquiries

1.2.2.2 AI Based Chatbot

In contrast to rule-based chatbots, AI-based chatbots are powered by artificial intelligence and can comprehend the context and intent of a users inquiry before generating a response. These chatbots use natural language processing to create their own solutions to complex questions and continuously improve their performance as they learn from the data they collect. AI chatbots are often considered as a more sophisticated version of chatbots and are particularly useful for businesses that deal with large volumes of data. Although they require more time and effort to train initially, they can save a significant amount of time in the long run, making them a worthwhile investment for businesses. One of the key advantages of AI chatbots is their ability to learn from data. As they interact with users, they collect and analyze data to understand user behavior and patterns. This information is then used to continuously improve their performance and make more informed decisions. AI chatbots also have a broader range of decision-making skills compared to rule-based chatbots. They can understand more complex questions and generate more personalized responses based on the users specific needs. Additionally, AI chatbots can communicate in multiple languages, making them suitable for businesses with a global presence. Overall, AI-based chatbots are an excellent choice for businesses that require a more sophisticated and intelligent chatbot solution. While they may require more resources to develop and train, they offer significant advantages in terms of efficiency, performance, and scalability. The introduction of chatbots revolutionized the way we interact with technology. Initially created to mimic human conversation, chatbots have now evolved to serve various purposes. With the ad-

vancement in chatbot technology, it is now possible to use chatbots for information retrieval, decision-making, shopping assistance, and even as language partners. However, chatbots also have their limitations. One of the most common uses of chatbots is as an information retrieval tool. Chatbots can provide users with relevant information on specific topics, from interactive FAQs to helping users make decisions. For instance, the YPA chat- bot, used by British Telecoms Yellow Pages, can search its knowledge base to retrieve the address of a plumber with an emergency service. Similarly, chatbots like ALEX can help law students find essential legal information. Another popular use of chatbots is as a language partner. Chatbots can help lan- guage learners practice their skills at any time, and they can incorporate both text and speech to improve reading, writing, and listening abilities. However, chat- bots responses may be predictable, repetitive, or lacking memory of previous responses, which limits their usefulness for advanced learners. In the field of education, chatbots can act as an amplifier of teaching methods, but not a replacement. Teachers can use chatbots to address student questions and generate log files of the conversations to identify students strengths and weaknesses. However, few chatbots are currently in use in education, which may be due to the limited documentation of their capabilities. It is worth noting that there are two types of chatbots: rule-based chatbots and AI-based chatbots [5]. Rule-based chatbots follow a pre-programmed set of rules, while AI-based chatbots use machine learning to understand the context and in- tent of a question before generating a response [6]. AI-based chatbots learn from the data they receive, continuously improve their decision-making skills, under- stand patterns of behavior, and can understand multiple languages. Overall, chatbots have come a long way since their inception and have various applications in different fields. Rule-based chatbots are best suited for simple tasks, while AI-based chatbots are ideal for complex tasks that require understanding the context and intent of a question. Although chatbots have theirimitations, they can serve as an amplifier of teaching methods and language learning tools.

1.3 Application of Project

The proposed university enquiry chatbot model has numerous applications and advantages, which are outlined below:

- **Accurate Information:** With the help of the university enquiry chatbot,

students can obtain accurate information in a timely manner. The chatbot's ability to process natural language allows for easy communication with students, and it can quickly provide them with the right source of information.

- **Instant Response:** The university enquiry chatbot, like any other chatbot, provides an instant response to student queries. This means that students do not have to wait long periods for a response, which can help them save time and increase their productivity. The chatbot's ability to respond 24/7 is also a significant advantage, as it can be available to students outside of regular office hours.

- **AI-based Chatbot System:** The proposed chatbot model uses AI-based techniques to process natural language and provide responses to student queries. This makes it highly efficient and effective in handling large volumes of enquiries from students. The AI-based chatbot system can be used by universities and businesses alike to streamline their enquiry processes and reduce workload.

- **Reduced Workload:** The university enquiry chatbot can significantly reduce the workload of universities by automating routine enquiries. This means that universities do not need to hire additional staff to handle the volume of enquiries from students. Instead, the chatbot can take care of routine enquiries, allowing staff to focus on more complex issues.

In addition to the above, the university enquiry chatbot can also improve the overall student experience. With the chatbot's ability to provide accurate and instant responses [7], students are more likely to be satisfied with the level of service they receive. The chatbot can also reduce the number of errors in responses and provide consistent information to all students.
Overall, the proposed university enquiry chatbot model has several applications and advantages that can be leveraged by universities and businesses to improve their enquiry processes, reduce workload, and provide better service to their customers. As the use of chatbots continues to increase, it is likely that they will become an essential part of many organizations' operations.

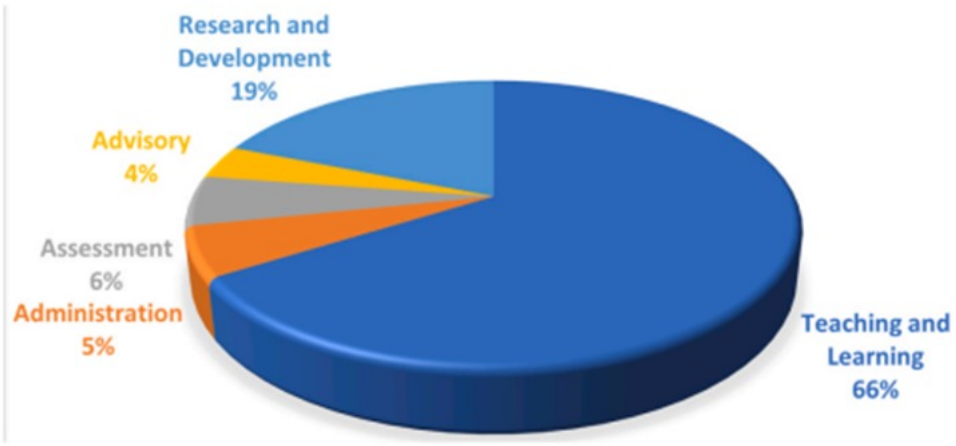


Figure 1.1: Uses of Chatbot in Education

1.4 Problem Statement

Artificial intelligence chatbot is an innovation that makes cooperations among man and machines utilizing normal language handling. We are building a chatbot for the school which answers the client questions with respect to the data about the school. The bot is coordinated to the Slack site. The bot answers to the client with fitting responses inside a negligible portion of second. It likewise contains investigation part which shows the quantity of clients who visited the site and used to bot to get the data about the University. This examination part should be visible simply by the administrator. People frequently get exhausted addressing a similar inquiry over and over and more over human can deal with all things considered 2-3 clients however by executing this bot, it can deal with n number of clients and it is accessible all opportunity in a day.

Chapter 2

LITERATURE REVIEW

2.1 Old Researches

A literature survey is an essential part of any research, providing a comprehensive overview of previous studies on a particular topic. In the field of education, chatbots have been gaining increasing attention due to their potential to enhance learning experiences by providing quick and efficient access to information. This chapter presents a review of the old research studies on chatbots in educational institutions. One such study was conducted by Dr. Vishwanath Karad [8] , who proposed a rule-based chatbot algorithm for an educational institution. The chatbot ana- lyzed user queries, matched them with a knowledge base, and provided appro- priate responses. While the study demonstrated the feasibility of chatbots in educational settings, it was limited to a rule-based algorithm and did not explore more advanced machine learning approaches. Another study by Harshala Gawade, Vedika Patil, Prachi Vishe, and Sonali Kolpe [9] utilized bigram and tokenization algorithms to build a chatbot for an educational institution. The chatbot analyzed user queries and used the bigram tokenization algorithm to identify key terms and provide relevant responses on a GUI screen. While this study demonstrated the potential of machine learning algorithms in chatbot development, it was limited to a specific set of algorithms and did not investigate the effectiveness of the chatbot in enhancing the learning experience. Mayuri Bhoir, Runa Ahirrao, and Madhuri Rodge [10] proposed a chatbot for an educational institution, which utilized the Chatterbot algorithm. The chatbot analyzed user queries and generated responses on a GUI screen. This study

demonstrated the effectiveness of the Chatterbot algorithm in developing chatbots for educational institutions, but it also had limitations, such as a lack of exploration of other machine learning algorithms. Overall, these previous studies demonstrated the feasibility of chatbots in educa- tional institutions and provided insights into the potential of various algorithms. However, they also had limitations, such as the use of limited algorithms and a lack of investigation into the effectiveness of the chatbots in enhancing the learn- ing experience. Future studies should explore more advanced machine learning algorithms and investigate the impact of chatbots on student learning outcomes.

Ref No.	Sr No.	Title	Year	Type	Framework	Language	Accuracy
[1]	1.	Chatbot developed for Educational Institute	2018	Rule Based Chatbot	Chatterbot	English	Not Mentioned
[2]	2.	College Enquiry Chat-Bot System	2020	Rule Based Chatbot	-------	English	Not Mentioned
[3]	3.	Web based College enquiry chatbot using .net	2020	Rule Based Chatbot	Bigram & Tokenization	English	Not Mentioned
	4	Self	2022	AI Based Chatbot	RASA Stack	English	TBD

Figure 2.1: Literature Review

2.2 Proposed System

The development of chatbots has transformed the way institutions interact with their stakeholders, especially in the education sector.the need for users to physically visit the institution for inquiries, as the chatbot can analyze queries and provide responses in real-time. The RASA Stack algorithm is used to perform calculations on the users queries and match them with the information stored in the database. The chatbot gener- ates an appropriate response that mimics a humans response to the query. The system utilizes a GUI interface that enhances the user experience, as the chatbot responses are presented in a manner that appears as if a real person is answering the inquiry. The University Enquiry Chatbot is designed to guide students and parents through the school enquiry process, providing relevant information with just a click on the chatbot. The proposed system also provides answers to general questions related to the school enquiry process. Users can raise any school-related activities through the chatbot, which responds with accurate information. The University Enquiry Chatbot is developed to enhance the overall experience of users in the institution and provide prompt solutions to their inquiries. Overall, the University Enquiry Chatbot is an innovative solution that simpli- fies the school enquiry process for administrators, students, and parents. The proposed system utilizes advanced algorithms to generate appropriate responses to user queries, thereby enhancing the efficiency and effectiveness of the en- quiry process. The GUI interface and natural language processing techniques employed by the chatbot provide a seamless experience for users, making the system user-friendly and easily accessible.the need for users to physically visit the institution for inquiries, as the chatbot can analyze queries and provide responses in real-time. The RASA Stack algorithm is used to perform calculations on the users queries and match them with the information stored in the database. The chatbot gener- ates an appropriate response that mimics a humans response to the query. The system utilizes a GUI interface that enhances the user experience, as the chatbot responses are presented in a manner that appears as if a real person is answering the inquiry. The University Enquiry Chatbot is designed to guide students and parents through the school enquiry process, providing relevant information with just a click on the chatbot. The proposed system also provides answers to general questions related to the school enquiry process. Users can raise any school-related activities through the chatbot, which responds with accurate information. The University Enquiry Chatbot is developed to enhance the overall experience of users in the

institution and provide prompt solutions to their inquiries. Overall, the University Enquiry Chatbot is an innovative solution that simpli- fies the school enquiry process for administrators, students, and parents. The proposed system utilizes advanced algorithms to generate appropriate responses to user queries, thereby enhancing the efficiency and effectiveness of the en- quiry process. The GUI interface and natural language processing techniques employed by the chatbot provide a seamless experience for users, making the system user-friendly and easily accessible.

Chapter 3

METHODOLOGY

The proposed system incorporates a combination of qualitative and quantitative research methodologies, which includes literature reviews, expert opinions, focus groups, and content validation. The system architecture includes an online enquiry module where students can enquire about various aspects related to their academic performance, examination schedules, fee structures, and placement opportunities. Given underneath is the framework design of this talk bot:

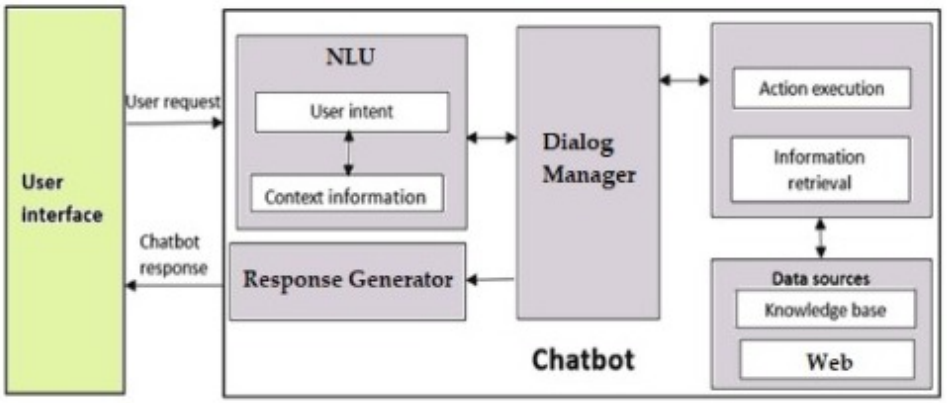


Figure 3.1: Methodology

The basic algorithm that will be implemented for the working of this proposed system is as follows:

1. **Step 1:** Start.
2. **Step 2:** Get the input query from the user.

3. **Step 3:** The query is pre-processed, extracting features such as intents and entities.
4. **Step 4:** Create a response using a dialogue manager.
5. **Step 5:** Match the fetched keywords with the keywords in the Knowledge base and perform the RASA Stack algorithm on it.
6. **Step 6:** Return the query response to the bot.
7. **Step 7:** Package the data into a proper response for display by the client.
8. **Step 8:** Exit.

Once the algorithm is performed, the chatbot returns the query response to the user. The chatbot then packages the data into a proper response for display by the client [11]. Finally, the system exits. This methodology ensures that the proposed chatbot system provides an effective and efficient solution to university enquiries, by enabling students to receive accurate and timely responses to their queries, without the need for physical presence or long waiting periods.
To further validate the proposed system's effectiveness, the research team will conduct a focus group study with a group of university students. This study will involve asking students to provide feedback on the chatbot's functionality, usability, and responsiveness. The data obtained from this study will be analyzed and used to refine the system, ensuring that it meets the requirements and needs of the target users. Overall, the proposed methodology is a robust and systematic approach to developing and evaluating the effectiveness of the university enquiry chatbot system.

3.1 Dataset Collection

In this section, we present the details of the dataset collection process for the development of a RASA Chatbot for university admission purposes. Our dataset comprises 74 unique scenarios, each containing 10 to 15 questionnaires. To ensure a comprehensive and diverse dataset, we employed a variety of sources, including:

- Internet research,
- Interviews with 12th-grade students,

- Social media groups such as WhatsApp and Facebook, and
- Personal networks.

We also leveraged Google Forms and interacted directly with students to obtain a rich and varied dataset. The dataset collection process involved multiple steps to ensure its accuracy and reliability. We initiated the process by conducting research on frequently asked questions related to university admissions, including admission criteria, academic requirements, and financial aid options. We then developed a struc- tured questionnaire with different questions for different departments and as- sisted support teams. We distributed these questionnaires to front desk staff, who assisted students in filling them out. We also gathered data from physical forms filled out at the universitys help desk, as well as through discussions and website visits. To ensure data con- sistency, we cross-checked the responses from different sources and eliminated duplicate entries. Additionally, we collected data from different departments, including the admissions office, academic departments, financial aid office, and student services. The collected dataset was then processed and cleaned to eliminate any outliers or incorrect responses. We employed various data cleaning techniques, including data transformation, removal of missing values, and outlier detection. In conclusion, the dataset collection process for the RASA Chatbot for univer- sity admission purposes involved multiple sources, methods, and techniques to ensure its accuracy, reliability, and comprehensiveness. The resulting dataset is a rich and diverse collection of questions and answers related to university admissions, which will serve as a valuable resource for the development of the chatbot.In this section, we present the details of the dataset collection process for the development of a RASA Chatbot for university admission purposes. Our dataset comprises 74 unique scenarios, each containing 10 to 15 questionnaires. To ensure a comprehensive and diverse dataset, we employed a variety of sources, including internet research, interviews with 12th-grade students, social media groups such as WhatsApp and Facebook, and personal networks. We also leveraged Google Forms and interacted directly with students to obtain a rich and varied dataset. The dataset collection process involved multiple steps to ensure its accuracy and reliability. We initiated the process by conducting research on frequently asked questions related to university admissions, including admission criteria, academic requirements, and financial aid options. We then developed a struc- tured questionnaire with different questions for different departments and as- sisted support teams. We distributed these questionnaires to front desk

staff, who assisted students in filling them out. We also gathered data from physical forms filled out at the universitys help desk, as well as through discussions and website visits. To ensure data con- sistency, we cross-checked the responses from different sources and eliminated duplicate entries. Additionally, we collected data from different departments, including the admissions office, academic departments, financial aid office, and student services. The collected dataset was then processed and cleaned to eliminate any outliers or incorrect responses. We employed various data cleaning techniques, including data transformation, removal of missing values, and outlier detection.

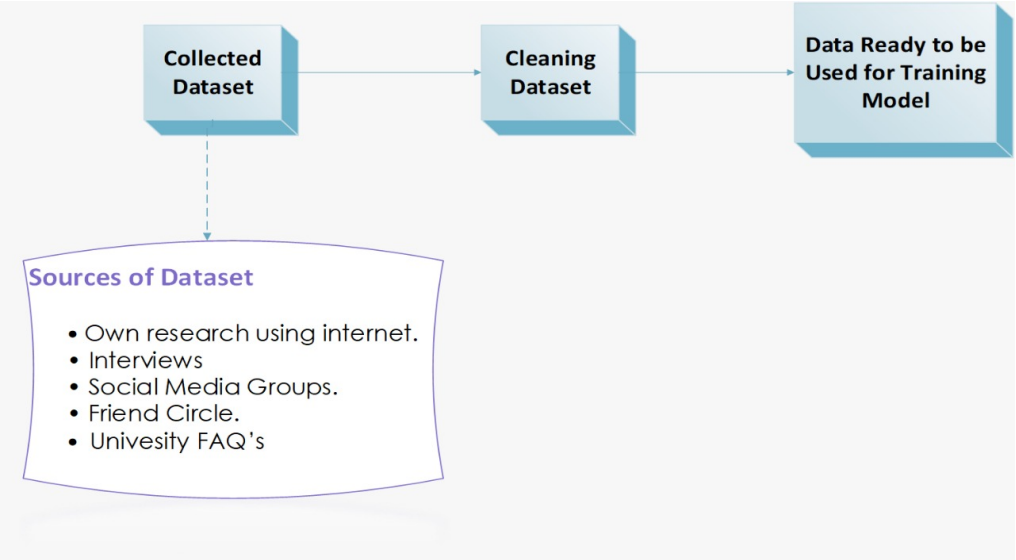


Figure 3.2: Dataset Collection

3.1.1 Examples

RASA is an open-source framework for building chatbots and virtual assistants [12]. It uses natural language processing (NLP) and machine learning (ML) to understand user input and provide relevant responses [13]. RASA chatbots can be integrated with various messaging platforms, including Facebook Messenger, WhatsApp, and Slack, making it an ideal solution for businesses and organizations that want to provide automated customer support or handle repetitive tasks. In the context of university admissions, a RASA chatbot can be used to answer student queries about admission procedures, program requirements, fee structures, and other relevant information. Now, lets take a closer look at some of the scenarios you collected in your dataset and how they can be handled by a RASA chatbot:

1. **Admission Schedule:** A RASA chatbot can be programmed to provide students with the latest information about admission schedules, deadlines, and important dates. It can also answer questions about the admission process and provide guidance on how to submit an application.

2. **Fee Structure:** A RASA chatbot can help students understand the different types of fees they need to pay, such as tuition fees, application fees, and other charges. It can also provide information about payment methods, deadlines, and any financial aid or scholarship programs that may be available.

3. **Offered Programs:** A RASA chatbot can assist students in selecting a program by providing information about the various programs offered by the university, including their requirements, curriculum, and duration. It can also answer questions related to program options, such as specializations or joint programs.

4. **Scholarship:** A RASA chatbot can help students understand the scholarship programs offered by the university and provide guidance on how to apply. It can also provide information about eligibility criteria, deadlines, and the application process.

5. **Credit Hour:** A RASA chatbot can provide students with information about credit hours required for different programs and courses. It can also provide guidance on how to calculate credit hours and how they affect a students academic progress.

6. **Hostel:** A RASA chatbot can provide information about hostel facilities and the application process. It can also answer questions related to hostel rules, fees, and other relevant information.

7. **Entry Test Pattern:** A RASA chatbot can help students understand the entry test pattern for different programs and provide guidance on how to prepare for the test.

8. **Evening Classes:** A RASA chatbot can provide information about evening classes offered by the university and answer questions related to their availability, program options, and other relevant information.

9. **Transfer:** A RASA chatbot can assist students in the transfer process by providing guidance on the requirements, documentation, and deadlines.

It can also answer questions related to eligibility, program options, and other relevant information.

10. **Events:** A RASA chatbot can provide information about university events, such as career fairs, seminars, and workshops. It can also assist students in registering for events and provide relevant information, such as the schedule and location.

11. **Aggregate:** A RASA chatbot can provide students with information about how the aggregate is calculated and how it affects their eligibility for different programs.

12. **Class Duration:** A RASA chatbot can provide information about class duration, such as the length of lectures and the number of classes per week.

13. **Summer Semester:** A RASA chatbot can provide information about summer semester programs offered by the university, including their duration, requirements, and eligibility criteria.

14. **Freezing Semester:** A RASA chatbot can help students understand the freezing semester process and provide guidance on how to apply for freezing a semester, including the required documentation, deadlines, and other relevant information.

15. **Campuses:** A RASA chatbot can provide information about different university campuses, including their location, facilities, and programs offered.

16. **Paper Rechecking:** A RASA chatbot can provide guidance on the paper rechecking process, including the requirements, fees, and deadlines. It can also answer questions related to eligibility and the outcome of the rechecking process.

17. **Mosque:** A RASA chatbot can provide information about the mosque facilities available on campus, including their location, prayer times, and other relevant information.

18. **Auditorium:** A RASA chatbot can provide information about auditorium facilities available on campus, including their location, capacity, and availability for events.

19. **Contacts:** A RASA chatbot can provide contact information for various university departments, including admissions, academics, student affairs, and other relevant departments.

20. **Address:** A RASA chatbot can provide the universitys address and location, as well as directions on how to get there.

Overall, a RASA chatbot can be a valuable tool for universities to provide students with quick and easy access to information related to admission procedures, program requirements, fee structures, and other relevant information. By leveraging natural language processing and machine learning, a RASA chatbot can provide personalized responses and assistance to students, helping them make informed decisions about their education and reducing the workload of university staff.

Figure 3.3: Examples of Dataset Senerios

3.2 Design Procedure

The process of designing a chatbot begins with understanding its fundamental components. First and foremost, the term "bot" refers to a computer program that performs automated tasks or executes specific commands based on

input data. In the context of chatbots, the program is designed to detect specific mes- sage patterns, either in the form of text or voice messages, submitted by users [14], and respond with pre-defined actions. Chatbots are capable of communicating through various channels, such as Face- book Messenger, Siri, WeChat, Telegram, SMS, Slack, Skype, and many others. Moreover, advancements in natural language processing and speech recognition technologies have enabled chatbots to respond to user inquiries with automated actions and voice messages, as demonstrated by Amazon Alexa, Google Home, and Microsoft Windows Cortana. It is worth noting that messaging apps have surpassed online entertainment plat- forms in terms of user engagement. This trend has led to an increased adoption of chatbot experiences by businesses and organizations, who use messaging apps as a primary channel for communicating with their customers. Therefore, designing a chatbot requires a deep understanding of the users needs and preferences, as well as the platform on which the chatbot will be deployed. The chatbots functionality and capabilities must align with the users require- ments, while also considering the limitations of the messaging platform. Ad- ditionally, the chatbots responses must be contextually appropriate, accurately reflecting the users intent and providing relevant information or services. Overall, the design of a chatbot requires a strategic approach that considers the fundamental components of the bot, the users needs and preferences, and the messaging platform on which the bot will be deployed. By aligning these ele- ments, a well-designed chatbot can provide an effective and efficient means of communication and service delivery for both businesses and users.

3.3 Chatbot Frameworks

Numerous business substances are attempting to make their own bots on the lookout, however when they decide to foster a bot, then they got befuddled between the stages for chatbot advancement. There are a lot of chatbot stages are accessible from which we can make chatbot, however it is challenging for specialty unit to choose. Beneath referenced are not many systems which are being utilized to make chatbot:

1. IBM Watson Conversation
2. Microsoft Bot Framework
3. Dialogflow

4. Amazon Lex
5. ManyChat
6. **RASA Stack**

We choose the **RASA Stack** system. asa stack contains a square of open source AI devices solely utilized in mean to make a relevant chatbots and collaborators. The administrations hold by this stage goes through a significant characterization of strong APIs and inserted along with Rasa stack which incorporates Rasa center and Rasa NLU as an occasion stream talked about all through this paper and furthermore the calculation engaged with expanding upon this stage. Its fixings incorporate the Bag of words calculation helping in working on portrayal utilized in the NLP, CRFs - Conditional Random Field utilized in measurable displaying and AI stages and furthermore trend setting innovation like LTSM brain organizations. It additionally supports the utilization of this RASA stage for the client required custom arrangement according to their necessities and furthermore elevates to contribute in fostering the stage for improved proficiency of the stage to work

3.4 Design Details

The RASA stack is one of the most broadly acclaimed open source visit bot building systems. The RASA NLU handles purpose acknowledgment and element revelation, though the RASA center handles setting and discourse the executives. Rasa NLU inside utilizes Bag of word calculation to track down goal and Conditional Random Field (CRF) to track down substances. The RASA center uses keras structure to execute a LTSM brain network for exchange the executives. This report targets giving an outline into the previously mentioned strategies which are utilized in RASA in the engine. This study was improved comprehend the understudy aide chatbot that we had worked to serve the University necessities.

3.4.1 RASA NLU Model

n this section, we will delve into the Rasa NLU model and its various components that aid in distinguishing goals and entities. The pipeline employed by Rasa involves parsing the text until it extracts the relevant entities and goals. The

process begins with text input, which is then processed through several stages to generate the desired output.

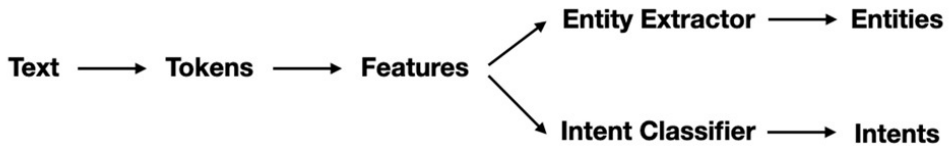


Figure 3.4: RASA NLU Model

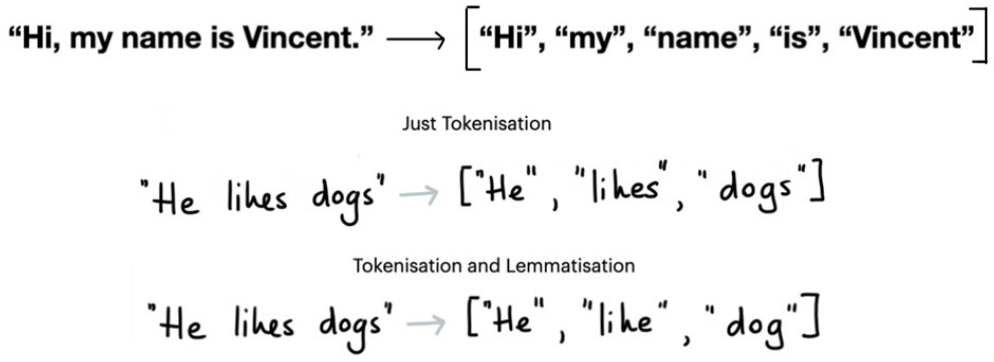
The Rasa pipeline includes various components, such as

1. Tokenizers
2. Featurizers
3. Intent Classifiers
4. Entity Extractors

that play a crucial role in the pipelines functioning. Let us discuss each of these components in detail

3.4.1.1 Tokenizers

The first step in the Rasa pipeline is to tokenize the input text into smaller pieces, known as tokens. This process occurs before the text is featurized for AI models, which is why the tokenizer is usually the first component in the pipeline.

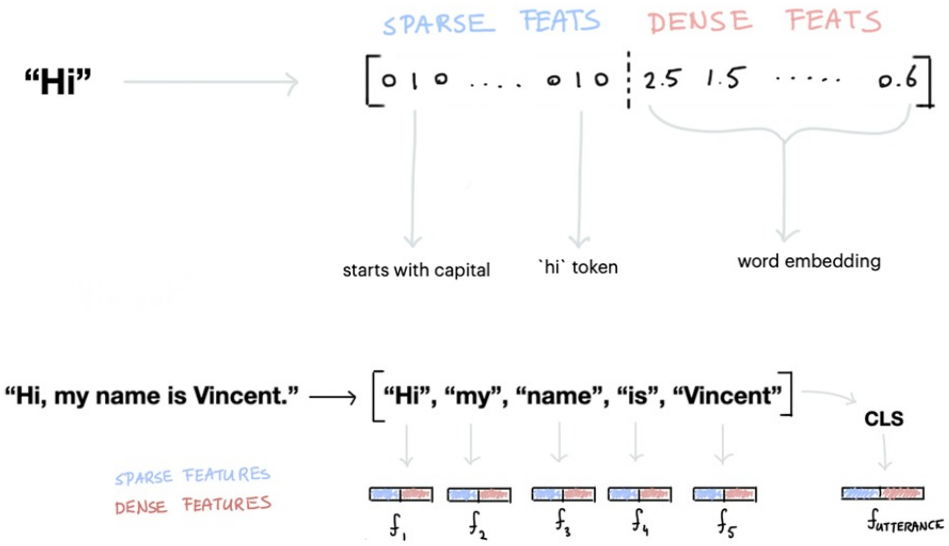


Tokenizers not only divide the text into tokens but also add additional information to each token. For example, spaCy can generate lemmas of the tokens that can be used by the CountVectorizer. The tokenizer splits each word in the text

into a separate token, resulting in a list of words. Tokens for punctuation marks may also be generated, depending on the tokenizer and the settings. For English, the preferred tokenizer is the WhiteSpaceTokenizer, while for non- English languages, other tokenizers such as spaCy and Jieba are commonly used. It is worth noting that tokenizers do not modify the underlying text; they only split the text into tokens. This means that capitalization remains intact, and lower-cased text encoding is the responsibility of the featurizer.

3.4.1.2 Featurizers

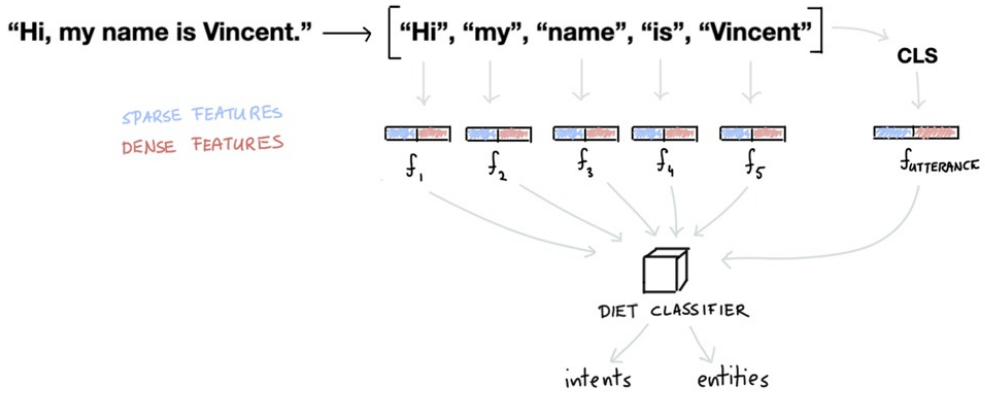
Featurizers create numeric highlights for AI models. The outline underneath shows how "Hey" may be encoded.



Featurizers are responsible for generating numeric features for AI models. They create features for tokens and the entire sentence, including the CLS token. The CLS token is a representation of the entire sentence, and its sparse features are a sum of all the sparse features of the tokens. The dense features, on the other hand, are either a pooled aggregate or mean of word vectors (in the case of spaCy) or a contextualized representation of the entire text (in the case of HuggingFace models). It is important to note that users can add their own components with custom featurization tools. For example, the rasa-nlu-models project maintained by the community contains several experimental featurizers for non-English languages. While it is not officially supported by Rasa, it can be beneficial to many users, given the numerous languages supported by the tool.

3.4.1.3 Intent Classifiers

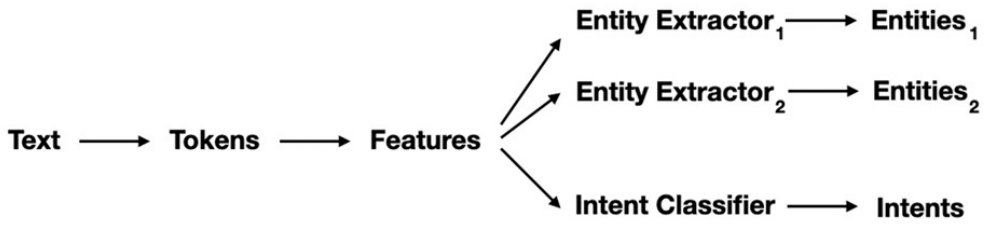
After generating features for tokens and the entire sentence, the next step in the pipeline is to pass the features to an intent classification model. We rec- ommend using Rasas DIET model, which can handle both intent classification and entity extraction [15]. It can also learn from both token and sentence features.



The DIET algorithm is unique in that it can perform both entity detection and intent classification, unlike most algorithms that Rasa previously hosted, which could only perform one of the two tasks. Furthermore, intent classification models in the past only considered the sentence features of the pipeline and ignored the token features. In conclusion, the Rasa pipeline involves several stages, including tokenization, featurization, and intent classification. Each of these components plays a critical role in the pipelines functioning and contributes to the successful extraction of entities and goals. By understanding the various components of the Rasa pipeline, developers can build more efficient and effective chatbots and virtual assistants.

3.4.1.4 Entity Extraction

Despite the fact that DIET is fit for figuring out how to recognize substances, we dont be guaranteed to suggest involving it for each kind of element out there. For instance, substances that follow an organized example, similar to telephone numbers, dont actually require a calculation to distinguish them. You can sim- ply deal with it with a RegexEntityExtractor all things considered. Therefore having more than one sort of substance extractor in the pipeline is normal.



Now that weve given an outline of the various sorts of parts in the NLU pipeline we can continue on to make sense of how these parts share data for one another. A characteristic language understanding arrangement which takes the client information and attempts to surmise the expectation and concentrate the accessible elements. The RASA NLU handles plan acknowledgment and substance disclosure. Rasa NLU inside utilizes Bag of word calculation to track down expectation and Conditional Random Field (CRF) to track down substances.

3.4.1.5 Steps for identifying features by RASA NLU

Input: First we take input.
Tokenization: Tokenize the input text.
Featurization: Make matix using:

- Sparse Featurizer
- Densed Featurizer

Transfomaion Layer: Extract intents and entities using Featurized Matrix.

3.4.1.6 Example:

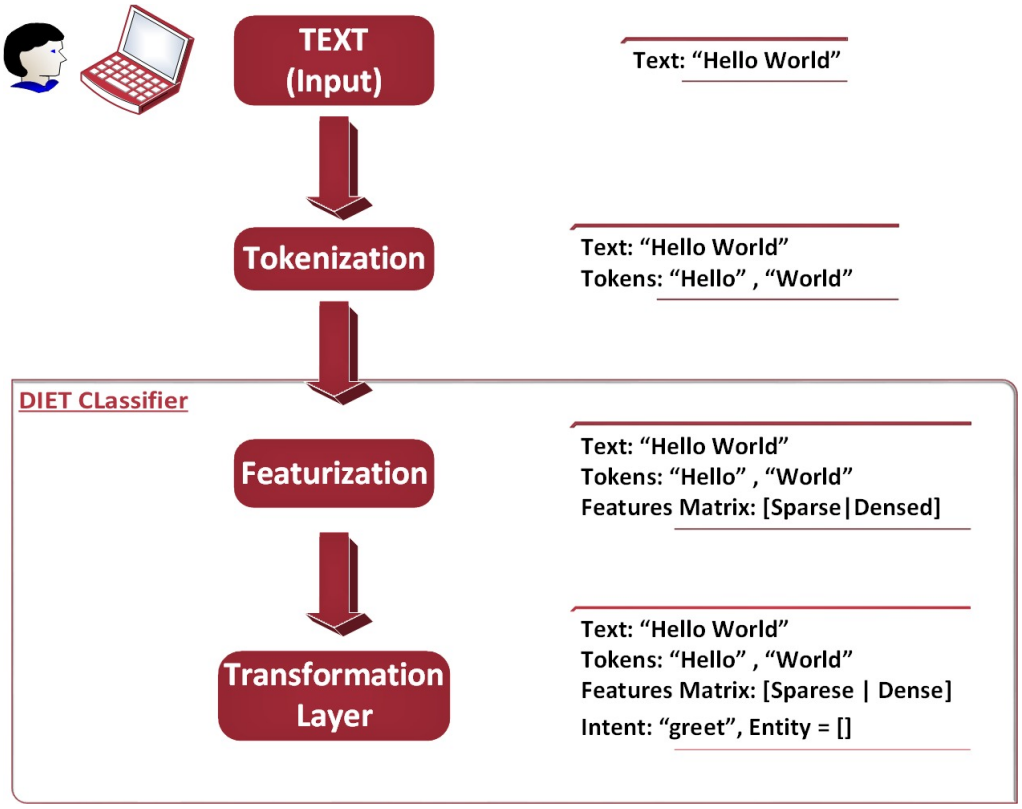


Figure 3.5: Internal Structure of RASA NLU

3.4.2 RASA Core

Rasa Core plays a crucial role in the Rasa framework as it handles the management of conversations between a chatbot and a user. One of its primary responsibilities is to track the context of the ongoing conversation and select the next appropriate action based on the current state of the conversation. This is accomplished by predicting the next probable "state" or goal of the chatbot conversation based on the submitted message. At runtime, Rasa Core relies on its specialized feature called "stories," which are sequences of potential goals submitted by engineers during the training stage. To execute conversation management, Rasa Core employs the Keras framework to implement a Long Short-Term Memory (LSTM) neural network. LSTM is a type of Recurrent Neural Network (RNN) [16], which has recurrent connections between its units or neurons. Unlike traditional neural networks, RNNs can process sequential data by maintaining information in their hidden state. This enables RNNs to analyze and

learn from input data that has temporal dependen- cies. An LSTM unit is a type of RNN unit that is specifically designed to learn long- term dependencies [17]. It has a unique architecture that allows it to store and access information for long periods, even when new data is introduced. LSTM units contain a memory cell, which is used to store the long-term memory of the network, and three gates that regulate the flow of information into and out of the memory cell [18]. These gates include the input gate, output gate, and forget gate, which are used to control the flow of information through the cell. When training the Rasa Core LSTM neural network, engineers submit "stories," which are sequences of potential goals or outcomes of a conversation. These stories are used to train the network to predict the next probable state or goal of the chatbot conversation based on the submitted message. The Rasa Core LSTM neural network processes the input message through a series of LSTM units or layers, each of which receives input from the previous layer and passes output to the next layer. One of the benefits of using an LSTM neural network for conversation management is that it can handle both short-term and long-term dependencies in the conversation. This is important because chatbot conversations often involve complex interactions between multiple topics and concepts, which can be difficult to track without a neural network. The use of an LSTM neural network allows Rasa Core to provide more personalized and relevant responses to users, based on their previous interactions with the chatbot. In conclusion, Rasa Core is an essential component of the Rasa framework that is responsible for managing conversations between a chatbot and a user. It uses a specialized LSTM neural network to predict the next probable state or goal of the chatbot conversation based on the submitted message. This allows it to provide more personalized and relevant responses to users, making it a valuable tool for chatbot developers.

3.4.2.1 LSTMs are RNNs

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that contains cyclic associations. It is a refined version of a standard RNN unit and is composed of gates that control the flow of information through the unit. The fundamental distinction between an LSTM unit and a standard RNN unit is the presence of gates in the former that regulate the input into the unit, the output from the unit, and what the cell should forget. The LSTM architecture comprises input gates, forget gates, and output gates that are signified by i(t), ft,

and Ot, respectively. In addition, the architecture includes a cell, which is the primary component that a neuron of a vanilla RNN contains. The LSTM unit is more refined because of the presence of these gates, which help in better controlling the flow of information through the unit. The input gate, for instance, regulates the information that enters the unit, whereas the forget gate regulates what information the cell should forget. The output gate regulates the information that exits the unit.

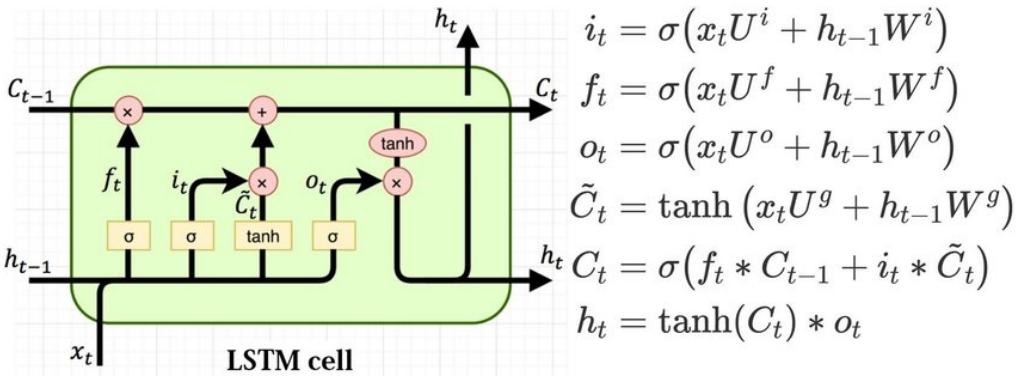


Figure 3.6: LSTM Artitecture

The LSTM architecture also contains cyclic associations, which allow it to process sequential information. The cyclic associations enable it to maintain information from previous time-steps and incorporate it into the current processing step. Any recurrent brain organization, including an LSTM, can be represented as a graph that contains one or more cyclic associations. The cyclic connections are critical to the LSTMs ability to handle sequential data effectively. The LSTM architecture is highly versatile and has several applications in natu- ral language processing, speech recognition, and other fields. To gain a deeper understanding of the architecture, one can refer to the original paper that pro- posed the LSTM by S. Hochreiter and J. Schmidhuber. Alternatively, there are numerous accessible resources online that provide an overview of the LSTM architecture and its components. In conclusion, the LSTM architecture is a type of RNN that contains cyclic associations and gates that regulate the flow of information through the unit. The architectures ability to handle sequential data and maintain information from previous time-steps makes it highly versatile and applicable to various fields. The LSTM architecture is a powerful tool for processing sequential data and has contributed significantly to advancements in natural language processing and speech recognition.

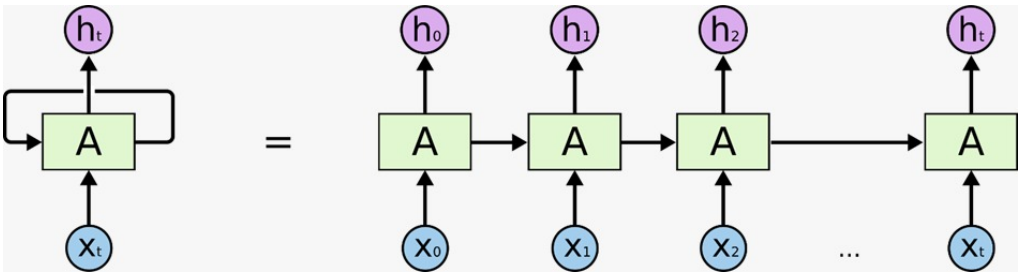


Figure 3.7: RNN Artitecture

3.4.2.2 Example

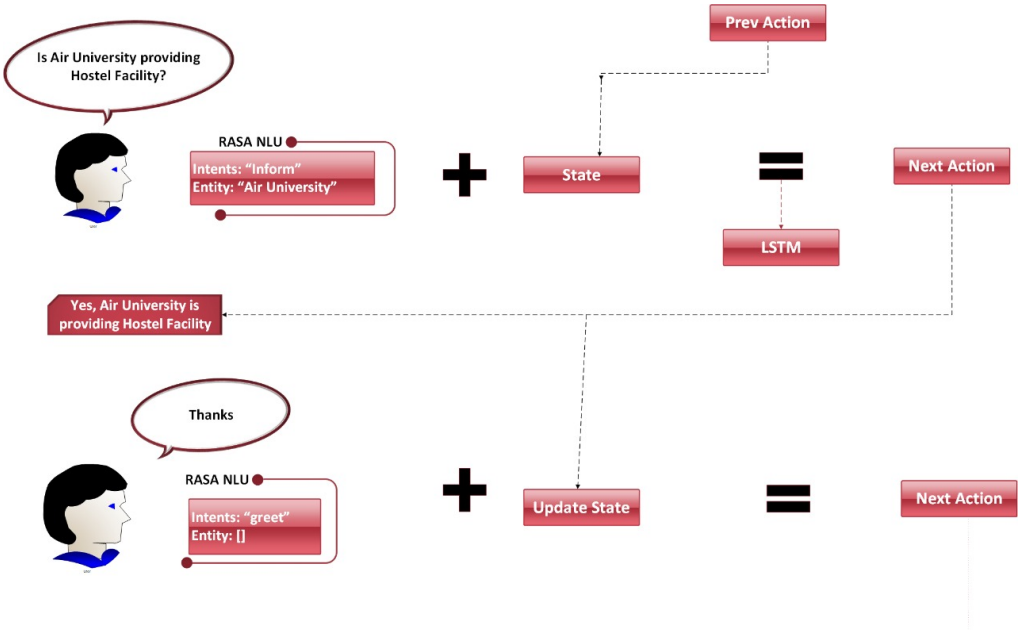


Figure 3.8: RASA NLU and RASA CORE Example

3.5 Integrating Chatbot with Slack and Facebook Page using Ngrok

To enable the chatbot to interact with users through Slack and Facebook Page, it is necessary to integrate the chatbot with these platforms. The integration process involves creating a bot account on each platform and configuring it to receive and respond to messages. In addition, a tool called Ngrok can be used to expose the chatbot running locally to the internet, allowing the chatbot to receive messages from Slack and Facebook. Ngrok is a lightweight tool that creates a secure tunnel between a local machine and the internet. By using Ngrok, the

chatbot can receive messages from Slack and Facebook even if it is running on a local machine. To use Ngrok, the chatbot needs to be running on the local machine and Ngrok needs to be installed and configured. Once Ngrok is installed and configured, the next step is to create a bot account on Slack and Facebook. For Slack, a bot account can be created through the Slack API website. For Facebook, a bot account can be created through the Facebook Developers website. Once the bot account is created, the bot token needs to be obtained and configured in the chatbot code. To enable the chatbot to receive messages from Slack and Facebook, the bot needs to be subscribed to the appropriate events. In Slack, this involves sub- scribing to the message event. In Facebook, this involves subscribing to the messages webhook. Once subscribed, the chatbot can receive messages from users and respond appropriately. In conclusion, integrating the chatbot with Slack and Facebook using ngrok en- ables the chatbot to interact with users through multiple channels. This enhances the user experience and increases the reach of the chatbot. By using Ngrok, the chatbot can be developed and tested locally before deploying it to a server, re- ducing development time and increasing productivity.

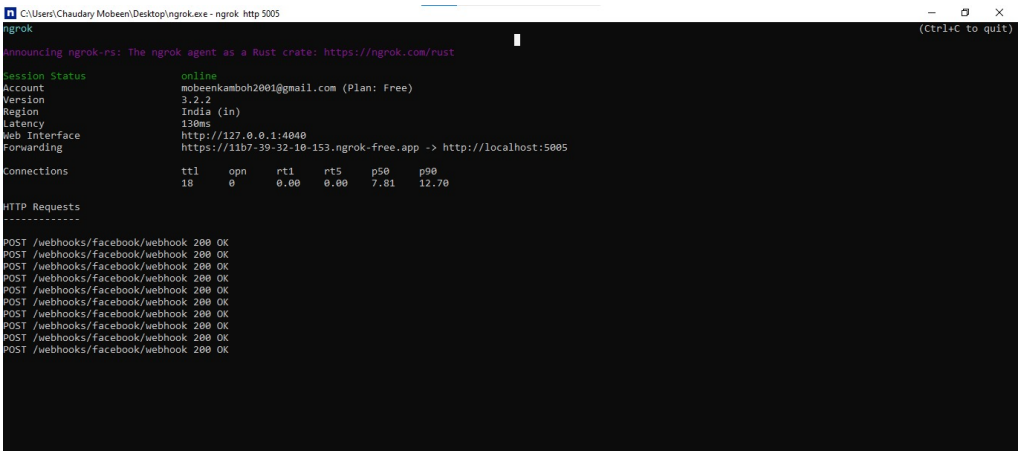


Figure 3.9: Connecting chatbot to Facebook and Slack using ngrok

3.6 Road Map of Project

Here is the road map of our whole project:

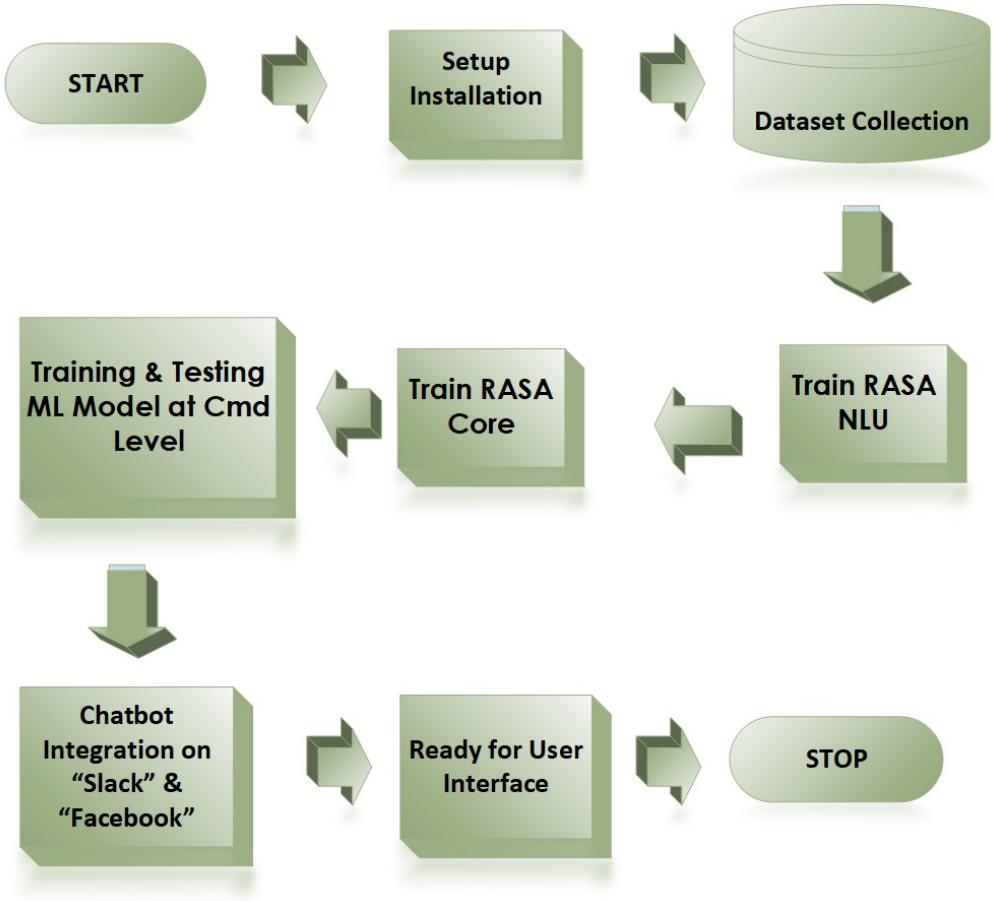


Figure 3.10: Block Diagram

Chapter 4

Results and Discussion

4.1 Results and Discussion

In this section, we present the results and discussion of our chatbot system. To illustrate how our model works, we provide an example scenario in which a user sends a message to the chatbot asking about the availability of hostel facilities at Air University. The message is first processed by the RASA NLU section, where intent and entities are extracted. These features are then passed to the RASA Core section, where a response is generated using features and RNN, and displayed on the screen.

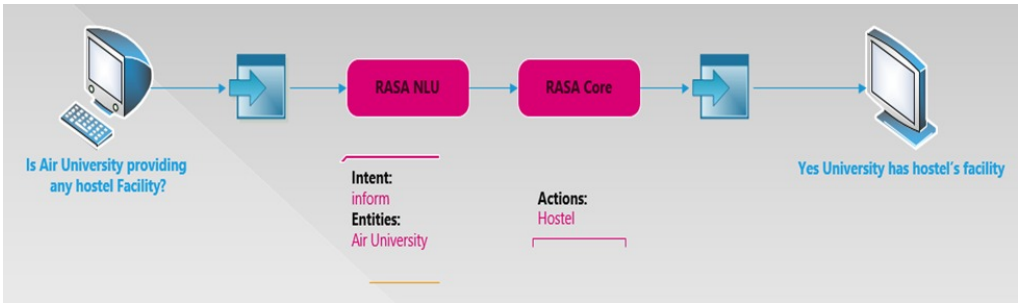


Figure 4.1: RASA Top View

To expound on the dialogue policy or dialogue management, our approach involves creating training data from stories and utilizing this data to train a model [19]. Stories refer to scenarios or schemes within a specific topic in any chat flow that we design or expect [20]. By creating training data from these stories, we are able to train our chatbot system to generate appropriate responses during a conversation. The dialogue policy is responsible for deciding which action to take

at each stage of the conversation. Different policies can be used and multiple policies can be included in a single agent or chatbot. The policy that predicts the next action with the highest confidence is used at each turn in the conversation. Our chatbot system employs two policies, namely the memoization policy and the neural network policy. The execution of the policy is based on priority, where the policy with the highest confidence score is executed first. The memoization policy records previous conversations in the training data and predicts the next action with a confidence score of 1.0 if the exact conversation exists in the training data. Otherwise, it predicts None with a confidence score of 0.0.
The neural network policy is implemented in Keras and selects the next action using a default architecture based on an LSTM. It is configured with a masking layer (input, 5, 32), LSTM units (32), dropout=0.2, and a dense layer (19) with 100 training epochs.
Regarding user interaction with the chatbot at the command-line level, this is a crucial aspect of the overall chatbot system. Users can interact with the chatbot by inputting text commands, which are then processed by the bot to provide relevant responses. This interaction is facilitated by the chatbot's natural language processing (NLP) capabilities, which enable it to understand user queries and generate appropriate responses. Users can provide feedback by rating the chatbot's responses, providing comments or suggestions, or reporting any errors or issues they encounter.
At the command prompt level, user interaction with the chatbot is facilitated by the chatbot's natural language processing (NLP) capabilities. Users can input text queries in natural language, and the chatbot utilizes algorithms and machine learning techniques to process these queries and provide relevant responses.
Providing feedback is an important aspect of user interaction with the chatbot. Feedback enables the chatbot to learn and improve its responses over time, leading to more accurate and useful responses in the future. Users can provide feedback by rating the chatbot's responses, providing comments or suggestions, or reporting any errors or issues they encounter.
In terms of the user interaction with the chatbot at the command-line level, the user can input text commands, which the bot then processes to provide relevant responses. This interaction is critical to the overall chatbot system's success [21]. The chatbot's ability to understand user queries and provide appropriate responses is essential to its effectiveness in addressing user needs.

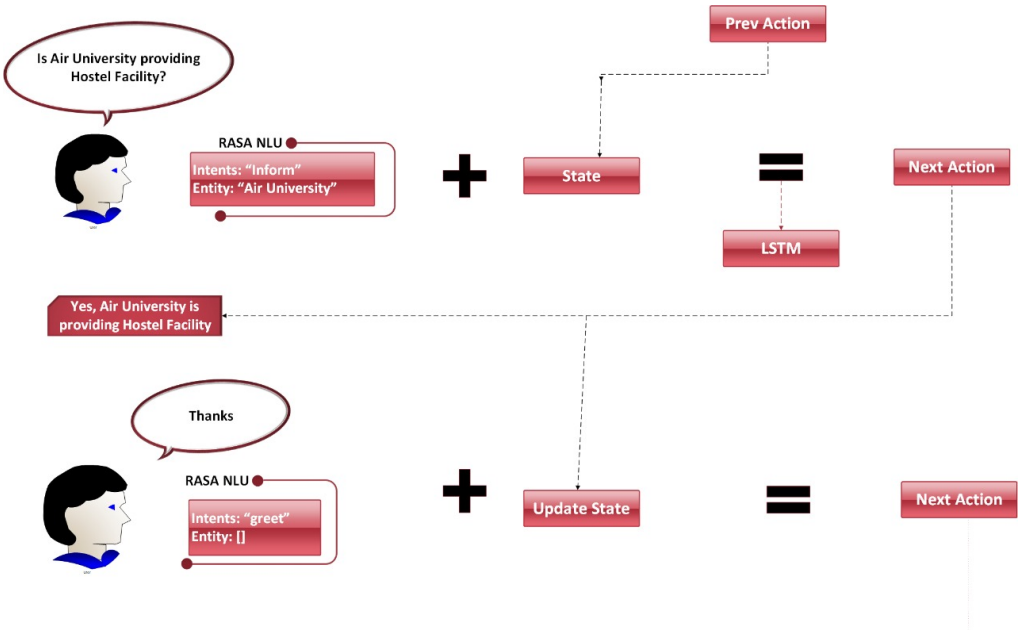


Figure 4.2: RASA NLU and RASA CORE Example

4.2 Interaction of Chatbot with User

4.2.1 Testing at Command Prompt Level

The user interaction with the chatbot at the command-line level is a crucial aspect of the overall chatbot system. At the command-line interface, users can easily interact with the chatbot by inputting text commands, which the bot then processes to provide relevant responses.

This interaction is facilitated by the chatbot's natural language processing (NLP) capabilities, which enable it to understand user queries and provide appropriate responses. Users can input text queries in natural language, and the chatbot utilizes algorithms and machine learning techniques to process these queries and provide relevant responses.

One important aspect of user interaction with the chatbot is the ability to provide feedback. Feedback enables the chatbot to learn and improve its responses over time, leading to more accurate and useful responses in the future. Users can provide feedback by rating the chatbot's responses, providing comments or suggestions, or reporting any errors or issues they encounter.User interaction with chatbot at cmd level is given by,

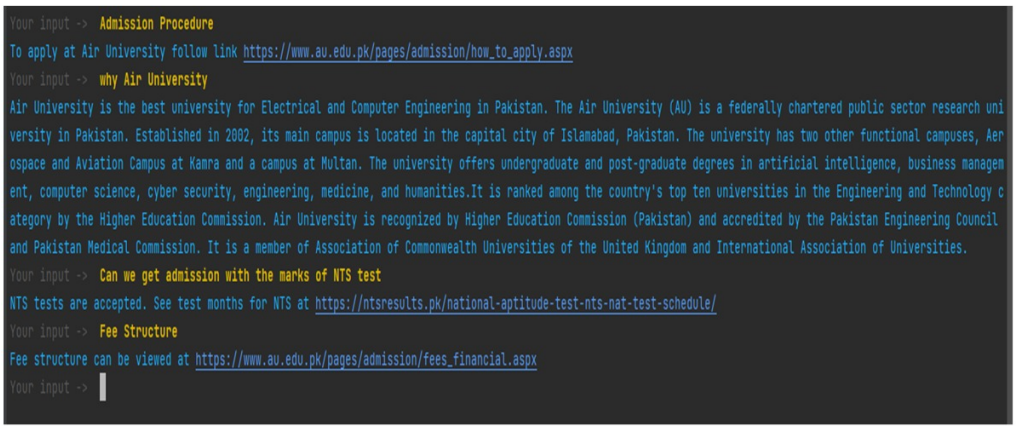


Figure 4.3: Interaction of Chatbot with User

4.2.2 Interaction on Slack

It is important to understand the user interaction with chatbots, and Slack is one of the platforms where chatbots are commonly used. When a user interacts with a chatbot on Slack, they can communicate through text messages, voice commands, or even interactive buttons.

Text messages are the most common way of interacting with a chatbot on Slack. The user types a message and sends it to the bot, and the bot processes the message using natural language processing (NLP) techniques. The bot then generates a response and sends it back to the user. The user can continue the conversation by typing another message. User interaction with chatbot at Slack is given by,

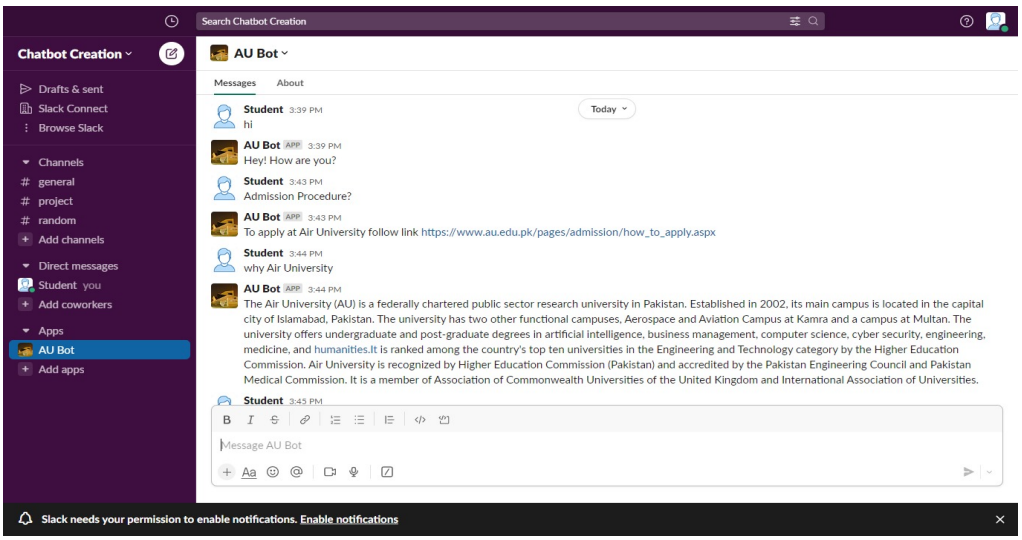


Figure 4.4: Interaction of Chatbot on Slack

4.2.3 Interaction on Facebook Page

The user interaction with the chatbot is a crucial aspect of the system. In the proposed University Enquiry Chatbot, the user interacts with the system through the Facebook page of the university. The chatbot is integrated into the Facebook Messenger platform, which enables users to send messages to the chatbot as if they are communicating with a human.

When a user sends a message to the chatbot, it triggers the chatbot to perform various tasks, such as retrieving information from the database, processing the user's input, and generating an appropriate response. The chatbot utilizes the RASA Stack algorithm to interpret and comprehend the user's message, which is then compared against the system's database to generate an appropriate response. The chatbot is capable of answering general inquiries about the university, such as admission criteria, course details, and other related queries.

The chatbot's ability to communicate with users via Facebook Messenger makes it easily accessible to students, faculty, and staff members. The chatbot's responses are generated quickly and accurately, ensuring that users receive timely and accurate information. The chatbot's integration with Facebook also makes it easy for the university to reach out to a wider audience and increase its online presence.

Overall, the user interaction with the chatbot is a seamless and convenient process that allows users to receive the information they need quickly and efficiently. The chatbot's ability to understand natural language and provide personalized responses makes it a valuable tool for the university's administration to provide quality customer service to its stakeholders.

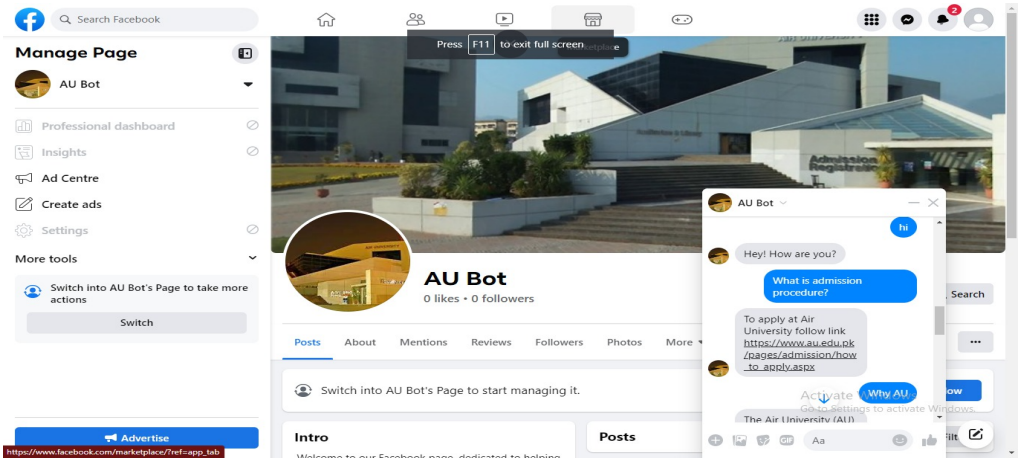


Figure 4.5: Interaction of Chatbot on Facebook Page

4.3 Accuracy and Precision Evaluation

The accuracy and precision of a chatbot model are crucial metrics to assess its performance and reliability. In the context of your Rasa chatbot, you have obtained valuable insights through the execution of the `rasa test` command. Let's delve into the results to analyze the accuracy and precision achieved by your model. Here is a screenshot of the evaluation results for the actions performed by the Rasa chatbot

```
2023-06-02 12:01:28 INFO     rasa.core.test  - Evaluation Results on ACTION level:
2023-06-02 12:01:28 INFO     rasa.core.test  -  Correct:          35 / 35
2023-06-02 12:01:28 INFO     rasa.core.test  -  F1-Score:        1.000
2023-06-02 12:01:28 INFO     rasa.core.test  -  Precision:       1.000
2023-06-02 12:01:28 INFO     rasa.core.test  -  Accuracy:        1.000
2023-06-02 12:01:28 INFO     rasa.core.test  -  In-data fraction: 0
2023-06-02 12:01:28 INFO     rasa.utils.plotting  - Confusion matrix, without normalization:
[[16  0  0  0  0  0  0]
 [ 0  3  0  0  0  0  0]
 [ 0  0  3  0  0  0  0]
 [ 0  0  0  4  0  0  0]
 [ 0  0  0  0  5  0  0]
 [ 0  0  0  0  0  3  0]
 [ 0  0  0  0  0  0  1]]
```

Figure 4.6: Evaluation Results on ACTION level

4.3.1 Accuracy

Accuracy is a fundamental evaluation metric that gauges the model's ability to correctly predict the outcome of conversations. It is calculated by dividing the number of accurately predicted conversations by the total number of conversations tested. In your case, the accuracy achieved is an astounding 100%. This indicates that your Rasa chatbot accurately predicted all 7 conversations during the testing phase. Such a high accuracy score underscores the effectiveness of your model in comprehending and responding to user inputs accurately.

4.3.2 Precision

Precision is an essential measure to evaluate the correctness of predicted actions by the model. It assesses the proportion of correctly predicted actions out of all the actions predicted by the model. In your evaluation results, the precision score achieved an impeccable 100%. This signifies that every single action predicted by your chatbot was correct, out of the 35 actions tested. This remarkable

precision score highlights the reliability and precision of your Rasa chatbot in generating appropriate responses and taking the intended actions.

The exceptional accuracy and precision exhibited by your Rasa chatbot indicate the high performance and reliability of the model. With an accuracy of 100%, your chatbot demonstrates a robust understanding of user conversations, enabling it to provide accurate and contextually relevant responses. The perfect precision score of 100% assures that every action executed by the chatbot aligns precisely with the desired outcome.

These outstanding evaluation metrics underscore the effectiveness of your Rasa chatbot in facilitating seamless and accurate interactions with users. The ability to achieve such high accuracy and precision scores indicates that your chatbot can confidently handle user queries and generate appropriate responses with a high degree of accuracy.

Chapter 5

Impact on Environment

The integration of AI-based chatbots in university admissions has the potential to create a positive impact on the environment. By leveraging these technologies, educational institutions can contribute to multiple Sustainable Development Goals (SDGs) related to education, gender equality, decent work, innovation, reduced inequalities, and strong institutions. This chapter will explore in detail how the implementation of AI-based chatbots for university admission aligns with these SDGs, highlighting their environmental implications and benefits.

5.1 Quality Education (SDG 4)

AI chatbots in university admissions facilitate access to inclusive and equitable quality education, reducing the need for extensive physical resources and paperwork. By providing prospective students with instant and accurate information about courses, admission requirements, scholarships, and educational opportunities, these chatbots promote efficiency in the admission process, leading to resource optimization and reduced environmental impact.

5.2 Gender Equality (SDG 5)

In the pursuit of gender equality, AI-based chatbots can address disparities in higher education. By providing information about initiatives and programs that promote gender equality in admissions, scholarships, and campus life, chatbots foster an inclusive environment for all genders. This inclusivity helps create a

diverse student body, encouraging collaboration and innovation to solve environmental challenges.

5.3 Decent Work and Economic Growth (SDG 8)

AI chatbots contribute to decent work and economic growth by aligning students' skills and interests with the job market. These chatbots provide valuable insights into career prospects, internship opportunities, and industry trends. By equipping students with relevant information, they help optimize career choices, potentially leading to sustainable employment and economic stability, while reducing mismatches between graduates' skills and job market needs.

5.4 Industry, Innovation, and Infrastructure (SDG 9)

The integration of AI chatbots in university admissions demonstrates a commitment to promoting innovation and developing efficient infrastructure. By streamlining inquiries, managing application data, and enhancing the admission process's efficiency, chatbots optimize resource utilization, reduce administrative burdens, and minimize the environmental footprint associated with traditional admission procedures.

5.5 Reduced Inequalities (SDG 10)

AI chatbots bridge information gaps and reduce inequalities in access to education. These chatbots provide a virtual assistant that ensures equal access to information and support for all prospective students, regardless of their geographical location or socioeconomic background. This inclusivity empowers a diverse range of students to pursue higher education, contributing to a more equitable society and fostering environmentally conscious leaders.

5.6 Peace, Justice, and Strong Institutions (SDG 16)

AI chatbots strengthen transparency and accountability in university admissions processes. By providing clear information about admission criteria, selection procedures, and timelines, these chatbots foster trust and integrity. Transparent and accountable admissions systems promote a fair and just environment, enhancing the reputation of educational institutions and ensuring that decisions are made based on merit. This, in turn, supports the development of environmentally responsible practices within the institution.

Let's conclude all,

The integration of AI-based chatbots for university admissions has a multifaceted impact on the environment, aligning with various SDGs. By leveraging these technologies, educational institutions can contribute to quality education, gender equality, decent work, innovation, reduced inequalities, and strong institutions, all of which have implications for environmental sustainability. The use of chatbots optimizes resource utilization, reduces administrative burdens, fosters inclusivity, and supports sustainable career choices. As educational institutions embrace AI-based chatbots, they demonstrate their commitment to creating a positive impact on the environment while advancing the SDGs.

Chapter 6

CONCLUSION

The development of a chatbot based on the RASA framework has provided a powerful tool for the University admission process. By leveraging the power of Artificial Intelligence (AI) and machine learning, the chatbot has been able to help 12th students stay updated on University admission activities. The RASA stack framework, which is built on top of Python and uses natural language processing (NLP), has enabled the chatbot to understand and respond to user queries in a more human-like manner.

RASA NLU (Natural Language Understanding) is a component of the RASA stack framework that is responsible for processing natural language input and extracting structured data. It is built on top of Python and uses machine learning algorithms to understand user input and classify it into intents and entities. This allows the chatbot to understand user queries and provide accurate responses.

RASA Core, on the other hand, is responsible for managing dialogue and conversation flow. It is also built on top of Python and uses machine learning algorithms to predict the next action to take based on the current conversation state. This allows the chatbot to have more natural and engaging conversations with users, as it can understand context and maintain the flow of the conversation.

In the development of the chatbot for University admission purposes, both RASA NLU and RASA Core were used to create a powerful and effective conversational agent. RASA NLU was used to process user input and extract intents and entities, while RASA Core was used to manage dialogue and conversation flow. The use of both components of the RASA stack framework allowed the chatbot to understand user queries and provide accurate and relevant responses, while maintaining a natural and engaging conversational flow.

The use of a knowledgeable database has allowed the chatbot to provide accurate

and relevant responses to user queries. The chatbot's ability to perform pattern matching and virtual assistance has made it possible to transform the user experience, making it more efficient and effective. With the help of the RASA stack framework, the chatbot has been able to reduce the workload on the University office staff, by automating the process of answering frequently asked questions. The chatbot has been designed to make conversation in English, providing a convenient and accessible way for students to get the information they need. The development of the chatbot involved the collection and preprocessing of a dataset, which was then used to train the RASA model. The interaction between the user and the chatbot was done at the command prompt level, providing a simple and intuitive interface for users to interact with the chatbot.

Overall the development of the RASA chatbot for University admission purposes has demonstrated the potential of AI and machine learning in transforming the user experience. The use of the RASA stack framework has enabled the chatbot to provide accurate and relevant responses to user queries, while reducing the workload on the University office staff. With further development and refinement, the chatbot has the potential to revolutionize the way universities interact with their students, providing a more efficient and effective means of communication.

Bibliography

[1] Master thesis halmstad university master's programme in information technology, 120 credits.

[2] Building a chatbotfor healthcare using nlp.

[3] Artificial intelligence in web development.

[4] Fb chatbot.

[5] Chatbots in customer service everything you need to know!

[6] Data science and machine learning in python.

[7] Pros and cons of chatgpt by openal tool [2023].

[8] Kshitija Shingte, Anuja Chaudhari, Aditee Patil, Anushree Chaudhari, and Sharmishta Desai. Chatbot development for educational institute. *Available at SSRN 3861241*, 2021.

[9] Harshala Gawade, Vedika Patil, Prachi Vishe, and Sonali Kolpe. College enquiry chat-bot system. *International Journal of Engineering Research & Technology*, 9(9):741–742, 2020.

[10] Rohan Parkar, Yash Payare, Keyur Mithari, Jitesh Nambiar, and Jaya Gupta. Ai and web-based interactive college enquiry chatbot. In *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–5. IEEE, 2021.

[11] Chatbot.

[12] Artificial intelligence chatbots.

[13] Artificial intelligence and user experience in reciprocity: Contributions and state of the art.

[14] Making of chatbot using rasa.

[15] The rasa blog.

[16] Intelligent log analysis for anomaly detection.

[17] Building a chatbot.

[18] Features of hearing: applications of machine learning to uncover the building blocks of hearing.

[19] Smart systems for universities.

[20] Developing fb chatbot based on deep learning using rasa framework.

[21] Chatbot ability.

Appendix A

Poject Code

A.1 config.yml

This file contains the configuration for the NLU (Natural Language Understanding) and Core components of the chatbot. It specifies the pipeline, language model, and other settings necessary for training and running the chatbot. A.6.

```
recipe: default.v1
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: 110
  constrain_similarities: true
- name: EntitySynonymMapper
- name: ResponseSelector
  epochs: 110
  constrain_similarities: true
- name: FallbackClassifier
```

```
  threshold: 0.3
  ambiguity_threshold: 0.1


policies:
- name: MemoizationPolicy
- name: RulePolicy
- name: UnexpecTEDIntentPolicy
  max_history: 5
  epochs: 110
- name: TEDPolicy
  max_history: 5
  epochs: 110
  constrain_similarities: true
assistant_id: 20230407-200143-scared-syrup
```

Listing A.1: 'Config.yml' Code

A.2 nlu.yml

This file contains training data for the NLU model. It includes examples of user messages and their corresponding intents and entities. This data is used to teach the chatbot how to understand user input. A.6.

```
version: "3.1"
nlu:
- intent: greet
  examples: |
    - hey
    - hello
    - hi
    - hello there
    - good morning
    - good evening
    - moin
    - hey there
    - lets go
    - hey dude
    - goodmorning
    - goodevening
```

```
    - good afternoon

- intent: goodbye
  examples: |
    - cu
    - good by
    - see you later
    - good night
    - bye
    - goodbye
    - have a nice day
    - see you around
    - bye bye
    - see you later

- intent: affirm
  examples: |
    - yes
    - y
    - indeed
    - of course
    - that sounds good
    - correct

- intent: deny
  examples: |
    - no
    - n
    - never
    - I dont think so
    - dont like that
    - no way
    - not really

- intent: mood_great
  examples: |
    - perfect
    - great
    - amazing
```

```
    - feeling like a king
    - wonderful
    - I am feeling very good
    - I am great
    - I am amazing
    - I am going to save the world
    - super stoked
    - extremely good
    - so so perfect
    - so good
    - so perfect

- intent: mood_unhappy
  examples: |
    - my day was horrible
    - I am sad
    - I dont feel very well
    - I am disappointed
    - super sad
    - Im so sad
    - sad
    - very sad
    - unhappy
    - not good
    - not very good
    - extremly sad
    - so saad
    - so sad

- intent: bot_challenge
  examples: |
    - are you a bot
    - are you a human
    - am I talking to a bot
    - am I talking to a human


- intent: scholarship
  examples: |
```

```
      - how can i get scholarship
      - criteria for scholarship
      - scholarship
      - financial aid
      - Can anyone guide me about scholarship
      - I need information about the scholarship procedure
      - I need guidance through the procedure of
        scholarship application
      - I need guidance for the application of scholarship
      - What is the process of scholarship application
      - I need information of the procedure of application
        of scholarship
      - what is the process of scholarship
      - how can I get scholarship
      - what is the process of financial aid
      - how can I get financial aid
      - what is the process of need based scholarship
      - what about study loan
      - how can I get study loan
      - how can I apply for study loan
      - how can I apply for financial aid
      - where can I apply for financial aid
      - where can I apply for scholarship

- intent: summer_warning
  examples: |
    - Can the warning count increase or decrease in summer
      semester
    - Will the warning count be changed in summer semester
    - does warning count is affected in summer semester
    - does warning count increment or decrement in summer
    - what about summer semester warning
    - Will the warning count be changed in summer semester
    - Can the warning count increase or decrease in summer
      semester
    - Will there be any warning count in summer
    - Will the warning count be incremented in summer
    - Will the warning count be changed in   semester
    - Can the warning count increase or decrease in summer
```

```
        semester

- intent: summer_courses
  examples: |
    - how many courses can we register in summer semester
    - how many courses can we register in summer
    - how many courses can we opt in summer semester
    - course limit in summer
    - courses in summer

- intent: AU
  examples: |
    - is Air University number 1 university
    - why should i choose Air University
    - why Air University is good for me
    - why should i apply at Air University
    - why Air University
    - AU
    - Air University
    - Air
    - why Air
    - why AU

- intent: job_rate
  examples: |
    - job rate at Air University in BCE
    - what is job rate of Air University
    - what is job rate at Air University
    - job rate
    -  job rate Air University
    -  job rate at Air University
    -  job rate at Air University
    -  job rate at Air University AU
    - what is job rate at Air University

- intent: admission_schedule
  examples: |
    - deadline for admission forms
    - admissions deadline
```

```
    - last date to apply
    - admission dates
    - schedule for admission dates
    - when can i apply for admission at Air University
    - when will test take place
    - when test will held
    - when should i apply for test
    - dates for apply
    - dates for admission
    - dates for entry test
    - dates of application submission
    - When can we apply for CS
    - What is the schedule with respect to the application
      of CS
    - When will the application of CS take place
    - What are the dates to apply for CS
    - What are the dates regarding the application of CS
    - What is last date of AU test
    - when is the last date of AU test
    - What would be the last date of the test of AU
    - What is the schedule for the AU Test
    - What is the information regarding the last date of
      AU test
    - What are the updates about the last date of AU test
    - What are the starting and ending dates of the
      admissions
    - When will the admission start and end
    - What is the information regarding the opening and
      closing date of admissions
    - What are the updates with regards to the starting
      and ending -date of admission
    - When will the admissions open and close
    - where result will be announced
    - when result will be announced for admission test


- intent: past_papers
  examples: |
    - where can I get sample test papers
```

```
202     - sample test papers
203     - past papers
204     - Do anyone have online material for the prepration of
          Air University admission test or anylink plz let
          me know if anyone have
205     - I need online preparation material for the Air
          University admission test
206     - What are the updates of the links to online material
          for the preparation of Air University admission
          test
207     - What is the information regarding the preparation of
          Air University admission test through online
          learning and links
208     - Online material and links for the preparation of Air
          University admission test are urgently needed I
          need information regarding this
209     - Online material for the preparation of Air
          University admission test is required What are the
          updates with regards to this
210     - Where can we get past papers of Air University
          regarding the entry test
211     - Where are the past paper of Air University entry
          test available
212     - Where can I get the past papers of Air University
          entry test
213     - Is there anyway I could find the past papers of Air
          University entry test
214     - Where would I be able to get a hold of the past
          paper of Air University entry test
215     - What is syllabus of entry test As I am an A level
          student
216     - What is the syllabus of entry test for an A levels
          student
217     - What would the syllabus of entry test be for a
          student who has cleared A levels
218     - I need information about the syllabus of entry test
          as I am an A levels student
219     - I have cleared my A levels what would the syllabus
          of entry test be for me
```

```
    - What syllabus to I have to go through for entry test
      being a student of A levels
    - How can a person prepare for the Air University
      entry
    - What are the ways to prepare for the Air University
      entry test
    - Does anyone know how to prepare for entry test of
      Air University
    - where can we get past papers of Air University
    - Where are the past paper of Air University available
    - Where will I be able to get past papers of Air
      University from
    - Is there anyway I could find the past papers of Air
      University
    - Where would I be able to get a hold of the past
      paper of Air University


- intent: admission
  examples: |
    - How can I apply in Air University if I have less
      than 60 marks in fsc
    - I have less than 60 in fsc How can I apply in Air
      University
    - What is the procedure of application in Air
      University for those students who have achieved
      less than 60 in fsc
    - What process should students with less than 60 go
      through to attain an admission in Air University
    - How can a student with less than 60 in fsc apply in
      Air University
    - Through which process can a student who has achieved
      less than 60 apply in Air University
    - I have completed my FSC in 2021 and unfortunately
      wasted an year after that Will I be able to apply
      on the basis of first year
    - Can I apply on the basis of first year eventhough I
      cleared my FSC in 2021 and ommited an year after
```

```
      that
242     - I had to miss an year after clearance of my FSC in
        2021 Will the university permit me to apply on the
        basis of my first year
243     - Will I be allowed to apply on the basis of first
        year despite the fact that I have ommited an year
        after the clearance of my FSC in 2021
244     - I cleared FSC in 2021 would I be permitted to apply
        on the basis of my first year eventhough I have
        missed an year
245     - Being an improver will I be able to apply for BSCS
246     - Will I be able to apply for BSCS as I am an improver
247     - Can an improver apply for admission in BSCS
248     - Is it possible for an improver to apply in BSCS
249     - Is it permissible for an improver to apply for
        admission in BSCS
250     - Is it possible to attain admission in BSCS in
        Islamabad if I have achieved 65 in HSSC-1 and 77
        marks in NTS
251     - I have attained 65 in HSSC-1 and 77 marks in NTS Is
        it possible to get an admission in BSCS in
        Islamabad
252     - Will it be possible for me to attain an admission in
         BSCS in Islamabaad if my HSSC-1 percentage is 65
        and NTS marks are 77
253     - With a result of HSSC-1 65 and 77 marks in NTS Is
        there anyway I could get an admission in BSCS in
        Islamabaad
254     - Can a student with an aggregate of 65 in HSSC-1 and
        77 marks in NTS get an admission in BSCS in
        Islamabaad
255     - what is the eligibility criteria for admission
256     - what is the selection criteria
257
258 - intent: admission2
259   examples: |
260     - Is it possible for a student to get admission even
        if he has D grades in A levels although the
        equivalence is above 60 or 70
```

```
261     - The equivalence is above 60 to 70 although the
          student has D grade in A levels Will he be eligible
           for an admission
262     - Having D grades in A levels along with the
          equivalence above 60 or 70 Will it be possible for
          the student to attain admission
263     - Is it possible to get an admission with an
          equivalence above 60 or 70 eventhough the grades in
           A levels are D
264     - The grades of A levels are D inspite of equivalence
          being above 60 or 70 Will it be possible for me to
          attain admission
265     - Will the student with lesser marks than 33 in a
          subject or in Intermediate pt1 be eligible
266     - Will the student with lesser marks than 33 in a
          subject or in Intermediate pt1 be eligible
267     - Is it possible for a student to get admission even
          if he has D grades in A levels although the
          equivalence is above 60 or 70
268     - The equivalence is above 60 to 70 although the
          student has D grade in A levels Will he be eligible
           for an admission
269     - Having D grades in A levels along with the
          equivalence above 60 or 70 Will it be possible for
          the student to attain admission
270     - Is it possible to get an admission with an
          equivalence above 60 or 70 eventhough the grades in
           A levels are D
271     - The grades of A levels are D inspite of equivalence
          being above 60 or 70 Will it be possible for me to
          attain admission
272     - What are the necessary requirements for a student of
           A levels
273     - Are there any essential requirements for an A levels
           student
274     - What will be required from a student who has cleared
           A levels
275     - What are the essentials for admission of an A levels
           student
```

```
276     - what kind of essentials would a student who has
          cleared A levels require
277
278
279 - intent: NTS
280   examples: |
281     - With NTS Test Can We Get Admission
282     - Can we get admission with the marks of NTS test
283     - Will I be able to attain admission using the marks
          of NTS test
284     - Is it possible to gain admission through the marks
          of NTS test
285     - Can NTS test marks help me in attaining admission
286     - Will it be possible to get an admission with the use
           of NTS test marks
287     - Can we apply with the marks of NTS test
288     - Will I be able to apply using the marks of NTS test
289     - Is it possible to apply through the marks of NTS
          test
290     - Can NTS test marks help me in applying
291     - Will it be possible to apply with the use of NTS
          test marks
292     - is NTS is compulsory for admission
293     - Is the test of NTS mandatory to get an admission
294     - Is it necessary to clear the NTS test for an
          admission
295     - To attain an admission Is it compulsory to clear a
          NTS test
296     - Is the NTS test essential for getting an admission
297     - For getting an admission Is it mandatory to pass a
          NTS test
298
299
300 - intent: NAT
301   examples: |
302     - I would like to inform that NAT result will be
          announced at the end of July and I wanted to ask
          whether we can apply on the basis of NAT test or
          not
```

```
303     - Will I be able to apply on the basis of NAT test
          The result of NAT test will be declared at the end
          of July
304     - The result of NAT test is going to come out at the
          end of July Furthermore Is it possible to apply on
          the basis of NAT test
305     - Can NAT test marks help me in applying  We will get
          to know our results of NAT test in the end of  July
306     - Will it be possible to apply with the NAT test
          results  Also It will be declared in the last week
          of July
307     - I would like to inform that NAT result will be
          announced at the end of July and I wanted to ask
          whether we can apply on the basis of NAT test or
          not
308     - Will I be able to apply on the basis of NAT test
          The result of NAT test will be declared at the end
          of July
309     - The result of NAT test is going to come out at the
          end of July Furthermore Is it possible to apply on
          the basis of NAT test
310     - Can NAT test marks help me in applying  We will get
          to know our results of NAT test in the end of  July
311     - Will it be possible to apply with the NAT test
          results  Also It will be declared in the last week
          of July
312     - is the AU test online or will it be conducted in Air
           University
313     - Where will the AU test be conducted  online or in
          Air University
314     - Online or in Air University Where will the test be
          conducted
315     - Where is the test going to take place Online or in
          Air University
316     - Online or in Air University where is the test going
          to take place
317
318 - intent: tests_conducted
319   examples: |
```

```
320     - Which test is required to be taken in order to get
          an admission in Air University
321     - which test is conducted to get admission in Air
          University
322     - Which test is necessary to attain admission in Air
          University
323     - Which is the essential test for admission in Air
          University
324     - Which test does Air University conduct for its
          admissions
325     - Which test is required to be taken in order to get
          an admission in Air University
326     - which test is conducted to get admission in Air
          University
327     - Which test is necessary to attain admission in Air
          University
328     - Which is the essential test for admission in Air
          University
329     - Which test does Air University conduct for its
          admissions
330
331
332 - intent: apply
333   examples: |
334     - How can the students from Balochistan apply
335     - What are the ways through which the students of
          Baclochistan may apply
336     - Is there anyway through which the students of
          Balochistan can apply
337     - What is the procedure for the students from
          Balochistan to apply
338     - What is the process of applying for the students of
          Balochistan
339     - I need information on how to apply along with the
          online admission website
340     - I need the website for applying and I also want to
          know what the process of application is
341     - What is the procedure of applying and from which
          website do we have to apply
```

```
342     - On which website does the application take place and
            what is its procedure
343     - How should I apply and please tell me the website
            from which I should apply
344     - how can we get admission in bscs
345     - What is the procedure to get an admission in Air
            University
346     - What is the process of attaining an admission in Air
            University
347     - How can I attain an admission in Air University
348     - I need guidance towards the attainment of admission
            in Air University
349     - I need information on how to get an admission in Air
            University
350     - What is the information regarding admission in
            Masters degree
351     - What are the details of getting an admission in
            Masters
352     - What are the updates with regards to the attainment
            of admission in Masters
353     - Is there any information available about getting
            admission in Masters
354     - How will a student attain admission in Masters
355     - When to apply for tests and Form Fee
356     - When should I apply for tests and what is the
            required amount of form fees
357     - How much amount is required for form fees and when
            can one apply for tests
358     - I need information on the application of tests and
            the amount of form fees
359     - I need details with regard to the date of
            application of tests and the amount of form fees
360     - What are the date of application of tests and how
            much is the amount of form fees
361     - When to apply for tests and Form Fee
362     - When should I apply for tests and what is the
            required amount of form fees
363     - How much amount is required for form fees and when
            can one apply for tests
```

```
364     - I need information on the application of tests and
          the amount of form fees
365     - I need details with regard to the date of
          application of tests and the amount of form fees
366     - What are the date of application of tests and how
          much is the amount of form fees
367     - What is the procedure to get an admission in Air
          University
368     - What is the process of attaining an admission in Air
           University
369     - How can I attain an admission in Air University
370     - I need guidance towards the attainment of admission
          in Air University
371     - I need information on how to get an admission in Air
           University
372     - What is the information regarding admission in
          Masters degree
373     - What are the details of getting an admission in
          Masters
374     - What are the updates with regards to the attainment
          of admission in Masters
375     - Is there any information available about getting
          admission in Masters
376     - How will a student attain admission in Masters
377     - what is the admission process
378     - how can I apply for admission
379
380
381 - intent: fee_structure
382   examples: |
383     - what is the admission fees
384     - what is the fee structure
385     - what is the tuition fee
386     - I need information of the token fees of Air
          University
387     - how much token fees will we have to pay in Air
          University
388     - What amount is essential for the payment of token
          fees in Air University
```

```
    - How much amount is paid for the token fees in Air
      University
    - I need information on the amount required to be paid
      as token fees in Air University
    - What will the semester fees be If I appear on the
      merit list
    - What is the semester fees of the students who are on
      the merit list
    - What will my semester fees be after appearing on the
      merit list
    - How much will my semester fees be if I achieve
      appearing on the merit list
    - After appearing on the merit list what will my
      semester fees be
    - What is the fee structure of MBA
    - How much is the fees of MBA
    - What is the amount of fee submission for studying
      MBA
    - How much fees will one have to pay in order to study
      MBA
    - What is the information regarding the fees of MBA
    - What are the complete fee structure of MSCS
    - I need information on the complete fee structure of
      MSCS
    - I need to know the total fee structure of MSCS
    - What are details with regards to the total fee
      structure of MSCS
    - Is there any information on the complete fee
      structure of MSCS


- intent: offered_programs
  examples: |
    - What fields of Engineering is offered by Air
      University
    - information of Bachelor Program.
    - Engineering in Air University is divided into which
      different offered fields
    - Which fields of Engineering does Air University
```

```
      offer to its students
414     - What are the sections of Engineering that are
          offered by Air University
415     - What sections of Engineering does Air University
          offer
416     - which degree programs are being offered for
          undergraduate students
417     - which degree programs are being offered for graduate
           students
418     - which degree programs are being offered for post
          graduates
419     - which degree programs are being offered for phd
420
421 - intent: hostel
422   examples: |
423     - is there hostel available
424     - hostel for girls
425     - hostel facility
426     - oncampus hostel
427     - incampus hostel
428
429 - intent: test_pattern
430   examples: |
431     - what is the test pattern
432     - what is the test pattern for bs cs
433     - I have cleared FSC with chemistry and now I opt to
          apply for BSCS What will my entry test be from
          computer or chemistry
434     - Will my entry test for BSCS contain questions from
          Chemistry along with Computer as I have cleared my
          FSC with Chemistry
435     - Will the entry test of BSCS contain information only
           regarding computer even if I have cleared my FSC
          with Chemistry
436     - Chemistry or computer What will my entry test for
          BSCS contain if I have passed my FSC with chemistry
437     - Currently applying for BSCS Will my entry test have
          questions from Chemistry besides computer as I have
           passed my FSC with Chemistry
```

```
    - Are there any specific subject required for an
      admission in BSCS
    - Is there any necessary subjects required for an
      admission in BSCS
    - What are the necessary subject requirements for BSCS
    - Which subjects are required for an admission in BSCS
    - Which subjects are essential with regards to getting
      an admission in BSCS
    - Are there any specific subject required for an
      admission in BSCS
    - Is there any necessary subjects required for an
      admission in BSCS
    - What are the necessary subject requirements for BSCS
    - Which subjects are required for an admission in BSCS
    - Which subjects are essential with regards to getting
      an admission in BSCS
    - Does BSCS entry test contain MCQs from Physics
    - Are MCQs from Physics included in BSCS entry test
    - Are MCQs from physics added in BSCS entry test
    - Will the entry test of BSCS have MCQs from physics
    - What is the pattern of Air University Engineering
      entry test
    - I need information about the Air University
      Engineering pattern of entry test
    - I need information on the pattern of Air University
      Engineering entry test
    - Is it possible for someone to tell me about the Air
      University Engineering entry test
    - What are the details regarding the Air University
      Engineering entry test

- intent: negative_marking
  examples: |
    - What is the information regarding negative marking
    - What are the details about the negative marking
    - What are the updates in regards to the negative
      marking system
    - Is there any information available about the
      negative marking
```

```
464     - How will the negative marking take place
465
466
467 - intent: evening_classes
468   examples: |
469     - Evening class
470     - I need information regarding the evening classes
471     - What are the details about the classes held in the
          evening
472     - What are the updates on the classes that are held in
           the evening
473     - What are the updates about the evening classes
474     - Has any information been provided about the evening
          classes
```

Listing A.2: 'Nlu.yml' Code

A.3 stories.yml

This file contains conversation data in the form of story templates. Stories describe example conversations between the user and the chatbot and are used for training the dialogue management model. A.6.

```
1  version: "3.1"
2
3  stories:
4  - story: Fallback
5    steps:
6    - intent: fallback
7    - action: utter_default
8
9  - story: PEC
10   steps:
11   - intent: PEC
12   - action: utter_PEC
13
14 - story: happy path
15   steps:
16   - intent: greet
17   - action: utter_greet
```

```
  - intent: mood_great
  - action: utter_happy

- story: sad path 1
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: sad path 2
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: scholarships
  steps:
  - intent: scholarship
  - action: utter_scholar

- story: graduation time
  steps:
  - intent: graduation
  - action: utter_graduate


- story: summer warnings
  steps:
  - intent: summer_warning
  - action: utter_summer

```

```
- story: admission schedules
  steps:
  - intent: admission_schedule
  - action: utter_schedule

- story: summer courses
  steps:
  - intent: summer_courses
  - action: utter_summer_courses

- story: why Air University
  steps:
  - intent: AU
  - action: utter_AU

- story: job rate at Air University
  steps:
  - intent: job_rate
  - action: utter_job

- story: social eventsat Air University
  steps:
  - intent: social_events
  - action: utter_events

- story: past papers
  steps:
  - intent: past_papers
  - action: utter_sample

- story: eligibility criteria
  steps:
  - intent: admission
  - action: utter_recruit

- story: A eligibility criteria
  steps:
  - intent: admission2
  - action: utter_admins
```

```
96
97  - story: NTS
98    steps:
99    - intent: NTS
100   - action: utter_nts
101
102 - story: NAT
103   steps:
104   - intent: NAT
105   - action: utter_nat
106
107 - story: tests conducted
108   steps:
109   - intent: tests_conducted
110   - action: utter_tests
111
112 - story: apply
113   steps:
114   - intent: apply
115   - action: utter_apply
116
117 - story: fee structure
118   steps:
119   - intent: fee_structure
120   - action: utter_fee
121
122 - story: Trips
123   steps:
124   - intent: Trips
125   - action: utter_Trips
126
127 - story: offered programs
128   steps:
129   - intent: offered_programs
130   - action: utter_programs
131
132 - story: hostel
133   steps:
134   - intent: hostel
```

```
135   - action: utter_hostel
136
137 - story: test pattern
138   steps:
139   - intent: test_pattern
140   - action: utter_pattern
141
142 - story: negative marking
143   steps:
144   - intent: negative_marking
145   - action: utter_negative
146
147 - story: evening classes
148   steps:
149   - intent: evening_classes
150   - action: utter_evening
151
152 - story: sports seats
153   steps:
154   - intent: sports_seats
155   - action: utter_sports
156
157
158 - story: self finance
159   steps:
160   - intent: self_finance
161   - action: utter_self
162
163
164 - story: domicile
165   steps:
166   - intent: domicile
167   - action: utter_domicile
168
169 - story: aggregate
170   steps:
171   - intent: aggregate
172   - action: utter_aggregate
173
```

```
- story: academic period
  steps:
  - intent: academic_period
  - action: utter_period

- story: Warning
  steps:
  - intent: Warning
  - action: utter_Warning


- story: Fyp Registration
  steps:
  - intent: Fyp_Registration
  - action: utter_fyp


- story: Transfer
  steps:
  - intent: Transfer
  - action: utter_Transfer


- story: GPA W
  steps:
  - intent: GPA_W
  - action: utter_gpaw

- story: GPA I
  steps:
  - intent: GPA_I
  - action: utter_gpai

- story: GPA
  steps:
  - intent: GPA
  - action: utter_gpa

- story: credit hour
```

```
213   steps:
214   - intent: credit_hour
215   - action: utter_credit
216
217 - story: classes
218   steps:
219   - intent: classes
220   - action: utter_class
221
222 - story: Admission of former
223   steps:
224   - intent: Admission_of_former
225   - action: utter_former
226
227 - story: awaited admission
228   steps:
229   - intent: awaited_admission
230   - action: utter_awaited
231
232 - story: semester week
233   steps:
234   - intent: semester_week
235   - action: utter_semester
236
237 - story: summer campus
238   steps:
239   - intent: summer_campus
240   - action: utter_campus
241
242 - story: summer withdraw
243   steps:
244   - intent: summer_withdraw
245   - action: utter_sum
246
247 - story: new summer
248   steps:
249   - intent: new_summer
250   - action: utter_new
251
```

```
- story: improve gpa
  steps:
  - intent: improve_gpa
  - action: utter_improve

- story: freezing semester
  steps:
  - intent: freezing_semester
  - action: utter_freezing

- story: Rechecking paper
  steps:
  - intent: Rechecking_paper
  - action: utter_Rechecking

- story: degree duration
  steps:
  - intent: degree_duration
  - action: utter_degree

- story: spring admission
  steps:
  - intent: spring_admission
  - action: utter_spring

- story: withdraw course
  steps:
  - intent: withdraw_course
  - action: utter_withdraw

- story: medals
  steps:
  - intent: medals
  - action: utter_medal


- story: career office
  steps:
  - intent: career_office
```

```
291   - action: utter_career
292
293 - story: dean
294   steps:
295   - intent: dean
296   - action: utter_dean
297
298
299 - story: headquarter
300   steps:
301   - intent: headquarter
302   - action: utter_headquarter
303
304
305 - story: AU Kamra address
306   steps:
307   - intent: AU_Kamra_address
308   - action: utter_Kamra
309
310 - story: AU Multan address
311   steps:
312   - intent: AU_Mul_address
313   - action: utter_Mul
314
315 - story: AU peshawar address
316   steps:
317   - intent: AU_pesh_address
318   - action: utter_pesh
319
320
321 - story: AU faislabad address
322   steps:
323   - intent: AU_fsd_address
324   - action: utter_fsd
325
326
327 - story: AU islamabad address
328   steps:
329   - intent: AU_isb_address
```

```
330   - action: utter_isb
331
332
333 - story: library
334   steps:
335   - intent: library
336   - action: utter_library
337
338
339
340 - story: cafeteria
341   steps:
342   - intent: cafeteria
343   - action: utter_cafeteria
344
345 - story: auditorium
346   steps:
347   - intent: auditorium
348   - action: utter_auditorium
349
350 - story: bast campus
351   steps:
352   - intent: best_campus
353   - action: utter_bestcampus
354
355
356 - story: academic office
357   steps:
358   - intent: academic_office
359   - action: utter_academic
360
361 - story: admin office
362   steps:
363   - intent: admin_office
364   - action: utter_admin
365
366 - story: student affair office
367   steps:
368   - intent: students_affair
```

```
  - action: utter_students

- story: hod office
  steps:
  - intent: hod_office
  - action: utter_hod

- story: director office
  steps:
  - intent: director_office
  - action: utter_director

- story: contacts
  steps:
  - intent: contacts
  - action: utter_contacts

- story: say goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye

- story: bot challenge
  steps:
  - intent: bot_challenge
  - action: utter_iamabot

- story: age
  steps:
  - intent: age
  - action: utter_age

- story: Foreigner
  steps:
  - intent: Foreigner
  - action: utter_Foreigner

- story: Foreigner2
  steps:
```

```
408   - intent: Foreigner2
409   - action: utter_Foreigner2
410
411 - story: online_registration
412   steps:
413   - intent: online_registration
414   - action: utter_online_registration
415
416 - story: Admission_Fee_Atm
417   steps:
418   - intent: Admission_Fee_Atm
419   - action: utter_Admission_Fee_Atm
420
421 - story: Application_Refund_Fee
422   steps:
423   - intent: Application_Refund_Fee
424   - action: utter_Application_Refund_Fee
425
426 - story: Admission_Form
427   steps:
428   - intent: Admission_Form
429   - action: utter_Admission_Form
430
431 - story: Fill_Form
432   steps:
433   - intent: Fill_Form
434   - action: utter_Fill_Form
435
436 - story: doc_req
437   steps:
438   - intent: doc_req
439   - action: utter_doc_req
440
441 - story: mul_page
442   steps:
443   - intent: mul_page
444   - action: utter_mul_page
445
446 - story: more_ug
```

```
  steps:
  - intent: more_ug
  - action: utter_more_ug

- story: weekend_degree
  steps:
  - intent: weekend_degree
  - action: utter_weekend_degree

- story: eligibility_criteria
  steps:
  - intent: eligibility_criteria
  - action: utter_eligibility_criteria

- story: CS_Medical
  steps:
  - intent: CS_Medical
  - action: utter_CS_Medical

- story: ALevel
  steps:
  - intent: ALevel
  - action: utter_ALevel

- story: Submit_or_not
  steps:
  - intent: Submit_or_not
  - action: utter_Submit_or_not

- story: Fee_ver
  steps:
  - intent: Fee_ver
  - action: utter_Fee_ver

- story: admit_card-issue
  steps:
  - intent: admit_card-issue
  - action: utter_admit_card-issue

```

```
- story: test_time_Pattern
  steps:
  - intent: test_time_Pattern
  - action: utter_test_time_Pattern

- story: re_appear_same_day
  steps:
  - intent: re_appear_same_day
  - action: utter_re_appear_same_day

- story: date_conflict
  steps:
  - intent: date_conflict
  - action: utter_date_conflict


- story: test_NS
  steps:
  - intent: test_NS
  - action: utter_test_NS

- story: recheck_entry_exam
  steps:
  - intent: recheck_entry_exam
  - action: utter_recheck_entry_exam

- story: recal_agg
  steps:
  - intent: recal_agg
  - action: utter_recal_agg

- story: timetable
  steps:
  - intent: timetable
  - action: utter_timetable

- story: Orientation_Book
  steps:
  - intent: Orientation_Book
```

```
525   - action: utter_Orientation_Book
526
527 - story: map_loc
528   steps:
529   - intent: map_loc
530   - action: utter_map_loc
531
532 - story: name_work
533   steps:
534   - intent: name_work
535   - action: utter_name_work
536
537 - story: chg_cnic
538   steps:
539   - intent: chg_cnic
540   - action: utter_chg_cnic
541
542 - story: salam
543   steps:
544   - intent: salam
545   - action: utter_salam
546
547 - story: attendence
548   steps:
549   - intent: attendence
550   - action: utter_attendence
551
552 - story: atten_Policy
553   steps:
554   - intent: atten_Policy
555   - action: utter_atten_Policy
556
557 - story: Result
558   steps:
559   - intent: Result
560   - action: utter_Result
561
562 - story: advisor
563   steps:
```

```
564   - intent: advisor
565   - action: utter_advisor
566
567 - story: Uni_timing
568   steps:
569   - intent: Uni_timing
570   - action: utter_Uni_timing
571
572 - story: Mosque
573   steps:
574   - intent: Mosque
575   - action: utter_Mosque
576
577 - story: Green Area
578   steps:
579   - intent: Green Area
580   - action: utter_Green Area
581
582 - story: Cafeinfo
583   steps:
584   - intent: Cafeinfo
585   - action: utter_Cafeinfo
586
587 - story: restarea
588   steps:
589   - intent: restarea
590   - action: utter_restarea
591
592 - story: dresscode
593   steps:
594   - intent: dresscode
595   - action: utter_dresscode
596
597 - story: Parking
598   steps:
599   - intent: Parking
600   - action: utter_Parking
601
602 - story: tempfac
```

```
603   steps:
604   - intent: tempfac
605   - action: utter_tempfac
606
607 - story: projectorboard
608   steps:
609   - intent: projectorboard
610   - action: utter_projectorboard
611
612 - story: Librarytim
613   steps:
614   - intent: Librarytim
615   - action: utter_Librarytim
616
617 - story: Commonlab
618   steps:
619   - intent: Commonlab
620   - action: utter_Commonlab
621
622 - story: Doc
623   steps:
624   - intent: Doc
625   - action: utter_Doc
626
627 - story: pet
628   steps:
629   - intent: pet
630   - action: utter_pet
631
632 - story: Party
633   steps:
634   - intent: Party
635   - action: utter_Party
636
637 - story: Curriculum
638   steps:
639   - intent: Curriculum
640   - action: utter_Curriculum
641
```

```
- story: outsider
  steps:
  - intent: outsider
  - action: utter_outsider

- story: mail
  steps:
  - intent: mail
  - action: utter_mail
```

Listing A.3: 'Stories.yml' Code

A.4 endpoints.yml

This file defines the endpoints that the Rasa server should connect to. It includes the configuration for the NLU and Core models, as well as any custom action servers. A.6.

```
# This file contains the different endpoints your bot can
   use.

# Server where the models are pulled from.
# https://rasa.com/docs/rasa/model-storage#fetching-models
   -from-a-server

#models:
#  url: http://my-server.com/models/default_core@latest
#  wait_time_between_pulls:  10   # [optional](default:
   100)

# Server which runs your custom actions.
# https://rasa.com/docs/rasa/custom-actions

#action_endpoint:
#  url: "http://localhost:5055/webhook"

# Tracker store which is used to store the conversations.
# By default the conversations are stored in memory.
# https://rasa.com/docs/rasa/tracker-stores

```

```
#tracker_store:
#    type: redis
#    url: <host of the redis instance, e.g. localhost>
#    port: <port of your redis instance, usually 6379>
#    db: <number of your database within redis, e.g. 0>
#    password: <password used for authentication>
#    use_ssl: <whether or not the communication is encrypted, default false>

#tracker_store:
#    type: mongod
#    url: <url to your mongo instance, e.g. mongodb://localhost:27017>
#    db: <name of the db within your mongo instance, e.g. rasa>
#    username: <username used for authentication>
#    password: <password used for authentication>

# Event broker which all conversation events should be streamed to.
# https://rasa.com/docs/rasa/event-brokers

#event_broker:
#  url: localhost
#  username: username
#  password: password
#  queue: queue
```

Listing A.4: 'Endpoints.yml' Code

A.5 credentials.yml

This file contains the credentials required for the chatbot to connect and authenticate with external services, such as databases, APIs, or messaging platforms. It typically includes information such as usernames, passwords, access tokens, and other authentication details. A.6.

```
# This file contains the credentials for the voice & chat platforms
```

```
# which your bot is using.
# https://rasa.com/docs/rasa/messaging-and-voice-channels

rest:
#  # you don't need to provide anything here - this channel doesn't
#  # require any credentials


#facebook:
#  verify: "AUbot"
#  secret: "1a970a08045ff1b23d19a3c0ef56e132"
#  page-access-token: "EAASgRcaFxpwBANDTWSa0hLiuNPbtHDbyaDhgqNpSkmkq4sJqHiG9M6teeG5AXtUmT"

#facebook:
#  verify: "AUbot"
#  secret: "1a970a08045ff1b23d19a3c0ef56e132"
#  page-access-token: "EAAxE1nHi0HkBAAyfEGJo21RF0LVc9dZBnMeaZCJZBxiyC1arRiwHQbTIVjHyh5vER"

#aubot
facebook:
  verify: "AUbot"
  secret: "6ff129559b56a9cbe31fb18571218aea"
  page-access-token: "EAAxE1nHi0HkBAMD6VpHa8ZBJY4BOAlgrshuZAVkurZBqBKAsdohOckbs2VeBZBM"


slack:
  slack_token: "xoxb-5128684221269-5145355609009-0IZPbNCDw2HkUjTBW1YCKcjK"
  slack_signing_secret: "88c6d55674f7c8df4a63cfaf3928c904"

#socketio:
#  user_message_evt: <event name for user message>
```

```
#  bot_message_evt: <event name for bot messages>
#  session_persistence: <true/false>

#mattermost:
#  url: "https://<mattermost instance>/api/v4"
#  token: "<bot token>"
#  webhook_url: "<callback URL>"

# This entry is needed if you are using Rasa Enterprise.
   The entry represents credentials
# for the Rasa Enterprise "channel", i.e. Talk to your bot
    and Share with guest testers.
rasa:
  url: "http://localhost:5002/api"
```

Listing A.5: 'Credentials.yml' Code

A.6 rules.yml

This file contains rules that define the dialogue flow and behavior of the chatbot. Rules are used to create specific conversation paths and trigger actions based on user input or system conditions. They provide a way to define the chatbot's responses and actions in a more structured and deterministic manner compared to using machine learning models alone. A.6.

```
version: "3.1"

rules:

- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye

- rule: Say 'I am a bot' anytime the user challenges
  steps:
  - intent: bot_challenge
  - action: utter_iamabot

```

```
15 - rule: Handle fallback
16   steps:
17   - intent: fallback
18   - action: utter_default
```

Listing A.6: 'Rules.yml' Code

Appendix B

Plagrism Report

FYP-III_C1977

ORIGINALITY REPORT

10%	9%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	aack.au.edu.pk Internet Source	2%
2	gitlab.sliit.lk Internet Source	1%
3	Submitted to University of Central Lancashire Student Paper	1%
4	blog.rasa.com Internet Source	1%
5	Submitted to Higher Education Commission Pakistan Student Paper	<1%
6	www.ijert.org Internet Source	<1%
7	arxiv.org Internet Source	<1%
8	Submitted to University of Teesside Student Paper	<1%
9	chatbotslife.com Internet Source	<1%

10	nu.edu.pk Internet Source	<1%
11	Maria Virvou. "Artificial Intelligence and User Experience in reciprocity: Contributions and state of the art", Intelligent Decision Technologies, 2023 Publication	<1%
12	ijsrcseit.com Internet Source	<1%
13	codeclimate.com Internet Source	<1%
14	www.coursehero.com Internet Source	<1%
15	Bruce Kevin, Brikes Vlkin, Manju Nair. "BUILDING A CHATBOT FOR HEALTHCARE USING NLP", Institute of Electrical and Electronics Engineers (IEEE), 2023 Publication	<1%
16	www.ijres.org Internet Source	<1%
17	Submitted to Universiti Teknologi Petronas Student Paper	<1%
18	nust.edu.pk Internet Source	<1%
19	code84.com Internet Source	

		<1%
20	Submitted to King's College Student Paper	<1%
21	Submitted to University of Derby Student Paper	<1%
22	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
23	Submitted to London Metropolitan University Student Paper	<1%
24	en.wikipedia.org Internet Source	<1%
25	www.alexandria.unisg.ch Internet Source	<1%
26	Submitted to University of St. Gallen Student Paper	<1%
27	onlim.com Internet Source	<1%
28	doctorpenguin.com Internet Source	<1%
29	digitalcommons.georgiasouthern.edu Internet Source	<1%