Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

UNIVERSITY M'HAMED BOUGARA-BOUMERDES

Faculty of **T**echnology

Department of Electrical Systems Engineering

Master's thesis

Presented by

BOUGUELIA ANIS

Field of Study: Telecommunications

Option: Networks and Telecommunications

BUILDING AND IMPLEMENTING A CUSTOMER SERVICE CHATBOT BASED ON RULES AND MACHINE LEARNING APPROACHES

Defended on 13/07/2023 in front of the jury composed of:

BELKACEM	Samia	MCA	UMBB	Chair
MESSAOUDI	Noureddine	MCA	UMBB	Examiner
GUERBAI	Yasmine	MCA	UMBB	Superviser

Academic Year: 2022/2023

ACKNOLDGMENT

First and foremost, I would like to express my profound gratitude to

الله عز وجل

for His blessings and guidance throughout this project. It is through His grace that I have accomplished this significant milestone, and I am humbled by His unwavering support.

I would like to express our sincere gratitude to my esteemed supervisor, ***Dr. Y.GUERBAI****, whose exceptional dedication, encouragement, and unwavering belief in my abilities have played a vital role in my success. I am deeply thankful for her invaluable guidance, insightful feedback, and continuous motivation throughout this project, which turned it into what it is today.*

I extend too my sincere appreciation too, to the esteemed ***members of the jury****, for accepting to evaluate my work. Their expertise, dedication, and valuable feedbacks refines my project and elevates its quality.*

A special word of thanks goes to ***Mr. MESSAOUDI****, the head of the department, for his exceptional leadership and countless efforts in fostering an environment of innovation and excellence. His guidance and support have been invaluable throughout our academic journey.*

Lastly, I am grateful to everyone who believed in me and provided support, no matter how small. Your contributions have made a significant difference in my success. Thank you all.

Dedications

To the extraordinary individuals who have been instrumental in my project's success and provided unwavering support, I offer a heartfelt dedication.

First and foremost, I wish to express my deepest appreciation and admiration for ***my family****, particularly* ***my beloved Mother****.*

Her unwavering belief in me, her boundless love, and her invaluable aid have been the driving force behind my achievements. This work is a tribute to her unwavering support and serves as a testament to the immeasurable impact she has had on my life.

I extend my profound gratitude to my esteemed university colleagues specially ***Lynda, Zakaria, Imene, Riham, Rym , Thanina , Ilham and Anis****, who have stood by my side throughout my academic journey.*

Their help, camaraderie and intellectual contributions have been invaluable in shaping my ideas and fueling my passion for knowledge. Through the ups and downs, they have stood by me, providing laughter, solace, and a sense of belonging. This dedication is a testament to the deep bond we share and a recognition of their significant role in shaping my journey, and I am truly fortunate to have such inspiring individuals by my side.

I wish to express my deepest appreciation to all my friends.

Including ***MCK and VNEA*** *team members,* ***Abderrahim****,* ***Lyess*** *and* ***Oussama****. Their unwavering friendship has been a constant source of strength and motivation. Through the ups and downs, they have stood by me, providing laughter, solace, and a sense of belonging. This dedication is a testament to the deep bond we share and a recognition of their significant role in shaping my journey.*

To my family, colleagues, and friends, I am forever grateful for your unwavering support, encouragement, and belief in my abilities. This dedication is a small token of my appreciation for the immense impact you have had on my life and the role you have played in making this project a reality.

SUMMARY

Chapter II: Methods used

Chapter III: Building and implementation

Table of figures

Chapter I: On defining Chatbots

Chapter II: Methods used

Chapter III: Building and implementation

Acronyms

AGI	Artificial General Intelligence
AI	Artificial intelligence
API	Application Programming Interface
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DL	Deep Learning
JSON	JavaScript Object Notation
ML	Machine learning
MLP	Multilayer Perceptron
NER	Named Entity Recognition
NLG	Natural Language Generation
NLI	Natural Language Inference
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NLU	Natural Language Understanding
NN	Neural Networks
POS	Part-of-Speech
RNN	Recurrent Neural Network
SNN	Simulated Neural Network
UI	User Interface

ملخص

مع التطور السريع للذكاء الاصطناعي في السنوات الأخيرة وظهور قسميه الفرعيين (تعلم الآلة والتعلم العميق)، أصبحت روبوتات المحادثة التي تعمل بنظام الذكاء الاصطناعي أدوات فعالة في المجتمع الحديث، لا سيما مع دمجها في مجموعة متنوعة من المجالات بهدف تحسين خدمة العملاء وتعزيز العمليات التجارية بشكل عام.

في هذا المشروع، سنتعلم أساسيات روبوتات المحادثة من أجل بناء روبوت محادثة مدعوم الذكاء الاصطناعي ومنظم جيدا وسهل الاستخدام للعملاء لمساعدة المسافرين عن طريق حجز تذاكر القطار تلقائيا في محاولة لتعزيز خدمات العملاء في قطاع النقل.

كلمات مفتاحية: الذكاء الاصطناعي، التعلم العميق، تعلم الآلة، روبوتات المحادثة

Abstract

With the rapid development of Artificial Intelligence (AI) in recent years and the appearance of its two subsections (Machine learning and Deep learning), AI-powered chatbots have become effective tools in modern society, particularly with their integration in a variety of fields with the goal of improving customer service and boosting business operations overall.

In this project, we will learn the fundamentals of chatbots in order to build an AI-powered chatbot that is well-structured and user-friendly for customers to assist travelers by automatically booking train tickets in an effort to enhance customer services in the transportation sector.

Keywords: AI, Machine learning, Deep learning.

Résumé

Avec le développement rapide de l'intelligence artificielle (IA) ces dernières années et l'apparition de ses deux sous-sections (l’apprentissage automatique et l'apprentissage profond), les chatbots alimenté par l'IA sont devenus des outils efficaces dans la société moderne, en particulier avec leur intégration dans une variété de domaines dans le but d'améliorer le service à la clientèle et d'accroître les opérations commerciales dans son ensemble.

Dans ce projet, nous apprendrons les principes fondamentaux des chatbots afin de construire un chatbot alimenté par l’IA qui soit bien structuré et convivial pour les clients pour aider les voyageurs en réservant automatiquement des billets de train dans un effort pour améliorer les services à la clientèle dans le secteur des transports.

Mots Clés : IA, L’apprentissage automatique, L'apprentissage profond, Chatbot.

General introduction

General introduction

In recent years, the revolution of artificial intelligence (AI) has made a profound impact across various industries, transforming the way we live and work. Its integration has become increasingly necessary as businesses strive to optimize processes, enhance efficiency, and improve customer experiences. One area that has received significant attention is the domain of customer services, where companies are actively seeking innovative solutions to address the challenges faced in delivering high-quality services.

One common complaint voiced by users refers to the low of customer services, particularly in the context of ticket bookings. Customers often encounter frustrating experiences, including long waiting times, repetitive queries, and impersonal interactions. These issues have prompted companies to explore alternative approaches to enhance their customer services and establish a competitive edge in the market.

In response to this prevalent problem, this thesis aims to offer a solution by leveraging the power of artificial intelligence. Specifically, we propose the development of a chatbot designed to streamline the process of train ticket bookings. By combining rule-based and machine learning approaches, this chatbot will provide an intelligent and efficient interface for customers, enhancing their overall experience and satisfaction.

The implementation of this AI-powered chatbot holds significant promise for improving customer services and driving business growth. By automating routine tasks and providing personalized assistance, the chatbot will reduce the workload on human agents, allowing them to focus on addressing more complex inquiries and providing a higher level of service.

In the following chapters, we will explore the process of developing and deploying this chatbot for train ticket bookings. We will delve into the details of the rule-based and machine learning methodologies utilized, outlining the steps taken to ensure the chatbot's effectiveness and accuracy.

By the end of this thesis, we aim to demonstrate the transformative potential of AI-driven chatbots in revolutionizing customer services within the train industry. Through this exploration, we hope to contribute valuable insights that will guide future advancements and assist companies in delivering superior customer experiences.

Chapter I:

On defining Chatbots

I.1. Introduction

In this chapter, we are going to define our main project of building a chatbot, first we will define the artificial intelligence (AI) and its subsections, after that we are going to have a detailed explanation of an AI-powered agent named chatbot, its history, types and its architecture which means how most of chatbots work.

I.2. History and evolution of Chatbots

The history and evolution of chatbots is a testament to the progress made in artificial intelligence and natural language processing. The concept of chat bots can be traced back to the 1960s when ELIZA, created by Joseph Weizenbaum, emerged as one of the earliest chatbots. ELIZA used pattern matching to simulate conversations, primarily emulating a Rogerian psychotherapist.

In the 1970s, PARRY, developed by Kenneth Colby, simulated a person with paranoid schizophrenia. It engaged in text-based conversations and responded to prompts related to its delusions.

During the 2000s, rule-based chatbots gained popularity. These chatbots followed predefined rules and if-then logic to generate responses. They were commonly used for customer support and frequently asked questions systems. However, their abilities were limited, and they struggled to understand natural language or handle complex conversations.

The 2010s marked a significant shift in chatbot technology. With advances in natural language processing, chatbots became more sophisticated. AI-powered chatbots emerged, leveraging machine learning algorithms to analyze and understand user input. Companies like Apple (Siri), Google (Google Assistant), and Microsoft (Cortana) introduced virtual assistants with voice recognition and natural language understanding capabilities.

The integration of chatbots with messaging apps, such as Facebook Messenger and WhatsApp, further accelerated their adoption. Businesses began using chatbots for customer service, enabling 24/7 support and handling basic inquiries. NLP-enabled

chatbots improved user interactions by extracting meaning, detecting intent, and providing contextually relevant responses.

OpenAI's GPT-3 and GPT-4, introduced in 2020 and 2023, represented a breakthrough in large-scale language models. GPT-4 could generate highly coherent and contextually appropriate responses, pushing the boundaries of chatbot capabilities.

Today, chatbots continue to evolve. They are integrated into various industries, including healthcare, finance, e-commerce, and more. Chatbot platforms offer developers tools to build sophisticated conversational agents, leveraging AI, NLP, and machine learning techniques. Ongoing research and advances in natural language generation and understanding promise even more human-like and personalized chatbot experiences in the future.

I.3. Artificial intelligence

Artificial Intelligence (AI) refers to the development and implementation of computer systems that can perform tasks that typically require human intelligence. It is a multidisciplinary field that combines computer science, mathematics, statistics, and various other disciplines create intelligent systems capable of reasoning, learning, and problem-solving.

AI aims to replicate and simulate human intelligence in machines, enabling them to perceive, understand, reason, learn, and make decisions. These intelligent systems often employ techniques such as machine learning, natural language processing, computer vision, and robotics to analyze data, extract patterns, and make informed predictions or take actions [1].

AI can be categorized into two main types: Narrow AI and General AI. Narrow AI, also known as weak AI, focuses on specific tasks and is designed to excel in a limited domain. Examples include voice assistants, recommendation systems, and image recognition software. On the other hand, General AI, also referred to as strong AI or Artificial General Intelligence (AGI), is a theoretical concept aiming to build machines capable of understanding and performing any intellectual task that a human can do.

This technology has numerous applications across various industries, including healthcare, finance, transportation, manufacturing, and entertainment. It has the potential to revolutionize these sectors by automating processes, improving decision-making, enhancing productivity, and unlocking new possibilities for innovation [2].

I.4. Machine learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that focuses on the development and application of algorithms and statistical models that enable computer systems to learn from and make predictions or decisions based on data, without being explicitly programmed. It encompasses the study of algorithms and techniques that allow machines to automatically acquire knowledge, improve performance, and adapt to changing environments.

At its core, machine learning aims to develop computational models that can automatically learn and improve from experience, without being explicitly programmed for each specific task. It focuses on creating algorithms and methods that enable machines to analyze and interpret patterns, relationships, and structures within data, and utilize this knowledge to generalize and make accurate predictions or decisions on new, unseen data [3].

I.4.1. Machine learning types

ML algorithms can be broadly categorized into three main types:

I.4.1.1. Supervised Learning

In this type of learning, the algorithm is trained on labeled data, where each example is associated with a known target or output. The algorithm learns to map the input data to the corresponding output by identifying patterns and relationships in the training data. It can then make predictions or classifications on new, unseen data.

I.4.1.2. Unsupervised Learning

Here, the algorithm is exposed to unlabeled data and aims to discover inherent patterns, structures, or relationships within the data. The algorithm learns to group or cluster similar data points together, find hidden patterns, or reduce the dimensionality of the data without any predefined target variable.

I.4.1.3. Reinforcement Learning

This type of learning involves an agent that interacts with an environment and learns to take actions to maximize a reward signal. The agent receives feedback in the form of rewards or penalties based on its actions, and through trial and error, it learns to make decisions or select actions that lead to higher rewards over time [4].

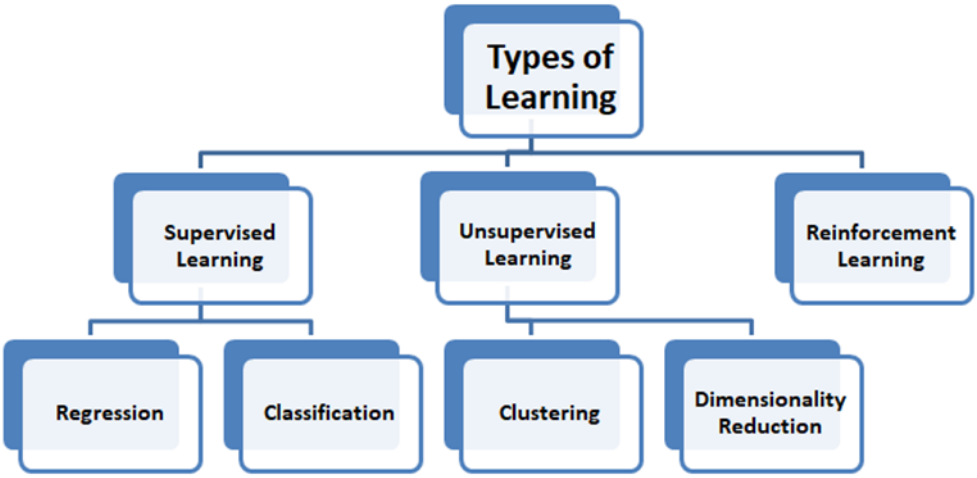

Figure I.1: Machine learning types [4].

I.4.2. Machine learning applications

Machine learning has wide-ranging applications in various fields, including image and speech recognition, natural language processing, recommendation systems, autonomous vehicles, fraud detection, medical diagnosis, and financial modeling, among others. It has the potential to extract valuable insights from large and complex datasets, automate decision-making processes, and enable intelligent systems that continuously learn and adapt to new information [3].

I.5. Deep learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy [5].

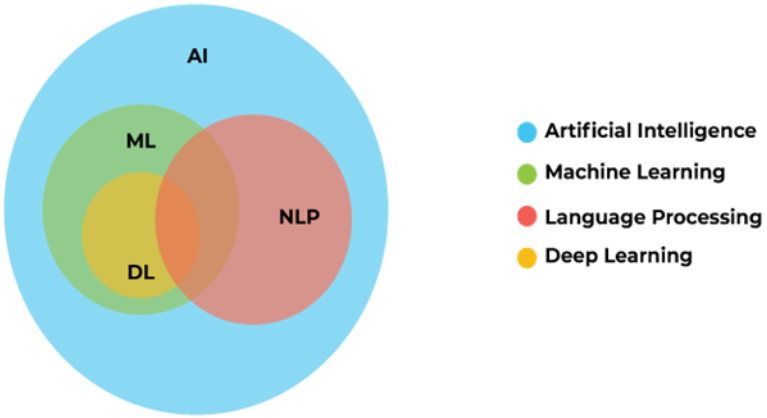

Figure I.2: Artificial intelligence and it's subsections [5].

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention [6]. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars) [W1].

I.6. Chatbots

I.6.1. Definition of a chatbot

A chatbot, also known as a conversational agent, is a computer program or AI system that utilizes natural language processing (NLP), artificial intelligence (AI), and machine learning (ML) techniques to simulate human-like conversation and interact with users. It serves as a virtual assistant or automated conversational interface that can communicate with users through text-based or voice-based channels [7].

Chatbots are designed to understand and interpret user input, whether in the form of text messages or spoken language, and generate appropriate responses based on the context and purpose of the interaction. They leverage NLP algorithms to analyze the structure and meaning of user queries, extract relevant information, and identify user intents and entities within the conversation [8].

The integration of AI and ML enables chatbots to continuously learn and improve their performance over time. Through ML algorithms, chatbots can process and analyze large volumes of data, including past conversations and user feedback, to gain insights and enhance their understanding of user preferences and behaviors. This iterative learning process allows

chatbots to provide more accurate and contextually relevant responses as they adapt to user needs [9].

Figure I.3: An AI powered agent, also called chatbot [7].

Chatbots can be categorized into two main types: rule-based chatbots and AI-powered chatbots. Rule-based chatbots follow predefined sets of rules and patterns to generate responses based on specific triggers or keywords. They are suitable for simple and structured interactions but may lack the ability to handle complex queries or understand nuanced language [10].

On the other hand, AI-powered chatbots leverage advanced AI techniques and ML models. These chatbots can handle a broader range of queries, learn from user interactions, and improve their performance over time. They can understand user intents, derive meaning from context, and generate more sophisticated and personalized responses [11].

Chatbots are widely deployed across various platforms and applications, including messaging apps, websites, mobile apps, and customer support systems. They serve numerous purposes, such as providing information, assisting with tasks or transactions, offering recommendations, and automating repetitive processes. Businesses often use chatbots to enhance customer support, streamline interactions, and provide round-the-clock assistance [12].

I.6.2. Types of chatbots

Chatbots are becoming increasingly popular in various industries, from customer service to healthcare. There are several types of chatbots, each with its own unique features and capabilities, theses chatbots are classified in six main types [13]:

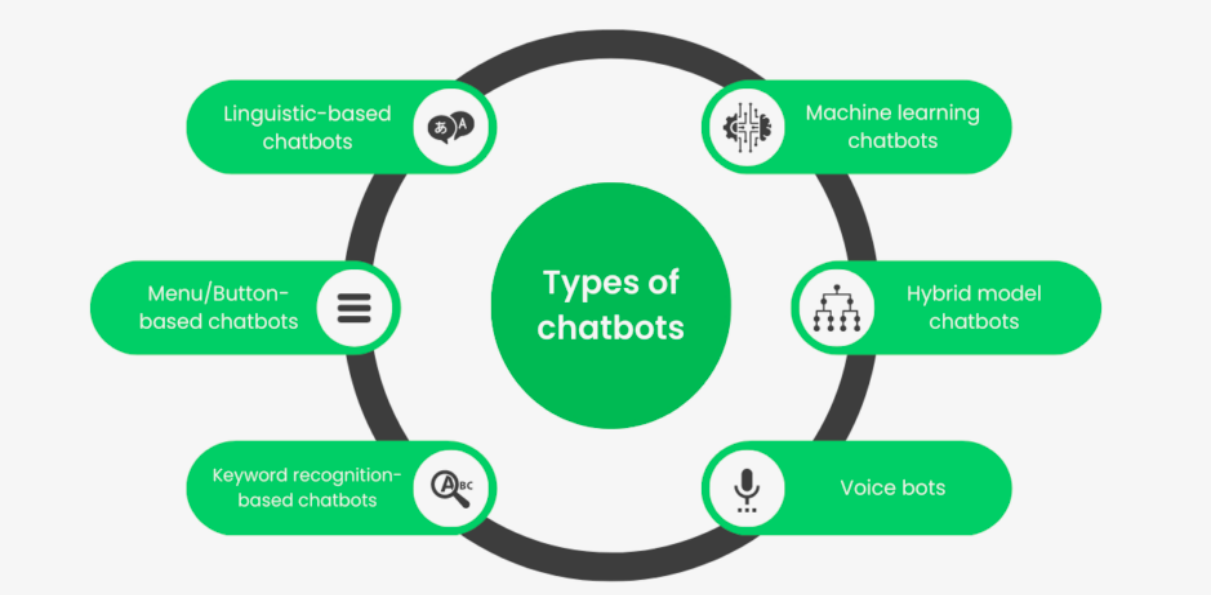

Figure I.4: Different types of chatbots [13].

I.6.2.1. Rule-based chatbots (Linguistic Based)

These chatbots are programmed to follow a set of predefined rules and can only respond to specific commands or keywords. They are typically used for simple tasks, such as answering frequently asked questions or providing basic customer support. Rule-based chatbots are easy to set up and maintain, but they are limited in their ability to understand natural language and provide personalized responses. These Rule-based chatbots use if/then logic to create conversational flows.

Language conditions can be created to look at the words, their order, synonyms, common ways to phrase a question and more, to ensure that questions with the same meaning receive the same answer. If something is not right in the understanding it's possible for a human to fine-tune the conditions.

However, chatbots based on a purely linguistic model can be rigid and slow to develop, due to this highly labor-intensive approach. Though these types of bots use Natural Language Processing, interactions with them are quite specific and structured. These type of chatbots tend to resemble interactive FAQs, and their capabilities are basic

I.6.2.2. AI-powered chatbots (ML & DL)

Chatbots powered by AI Software are more complex than rule-based chatbots and tend to be more conversational, data-driven and predictive. These type of chatbots is generally more sophisticated, interactive and personalized than other task-oriented chatbots. Over time with

data they are more contextually aware and leverage natural language understanding and apply predictive intelligence to personalize a user's experience. Conversational systems based on machine learning can be impressive if the problem at hand is well-matched to their capabilities. By its nature, it learns from patterns and previous experiences.

I.6.2.3. Hybrid chatbots

In Hybrid chatbots, rule-based and AI-powered approaches are combined together to provide more accurate and personalized responses. They are designed to provide the best of both technologies, with the simplicity of rule-based chatbots and the intelligence of AI-powered chatbots. Hybrid chatbots are typically used for complex tasks, such as customer service and sales. They require more development time and resources than rule-based chatbots, but they can provide a better user experience and improve customer satisfaction.

I.6.2.4. Menu/Button-based chatbots

Menu-based chatbots are the most basic types of chatbots implemented in the market today. These chatbots work on a decision tree logic whereby users seek answers to their queries by following a set of buttons.

This makes menu-based chatbots especially popular among businesses that need an efficient means of answering FAQs. Unfortunately, menu-based chatbots fall short when it comes to complex applications with too many variables. They are also the slowest chatbots available.

I.6.2.5. Voice-activated chatbots

These chatbots are designed to respond to voice commands and are commonly used in smart home devices and virtual assistants like Amazon's Alexa or Google Home. They are typically AI-powered and use natural language processing to understand voice commands. Voice-activated chatbots are designed to be user-friendly and provide a seamless experience for the user. They require integration with voice recognition software and may require additional development time and resources.

I.6.2.6. Keyword-recognition based chatbots

Keyword-recognition based chatbots are an extension of menu/button-based chatbots. With keyword recognition-based chatbots users can interact by giving free text input. The chatbot then analyses the text on specific keywords and based on those keywords, gives an appropriate response.

For example, when a user asks "What are the benefits of a car X?", the chatbot could recognize the keywords "benefits" and "car X", and give a list of possible benefits of using the specific car [14].

➢ Comparing between rules-based chatbots and AI chatbots

Rules-based chatbots are simple and have limited capabilities. They are programmed to recognize a keyword or phrase and deliver a canned response based on that input.

To avoid the frustration of the chatbot not understanding user input, many rules-based chatbots are "button-based" and guide users down a decision tree where they are able to choose between a limited amount of answers.

On the other side, AI-powered, NLP chatbots are far more sophisticated in their uses. AI chatbots use natural language processing (NLP) to determine the intent behind a user's question. Instead of relying on keywords or buttons, users can type as they would talk to a human agent and the bot can understand the context and respond accordingly.

AI chatbots can often resolve requests without human interaction and they learn and grow as time goes on. This is why they are often referred to as virtual agents or intelligent virtual assistants, because they can respond in a human-like way and can resolve certain requests, especially simple or repetitive ones [20].

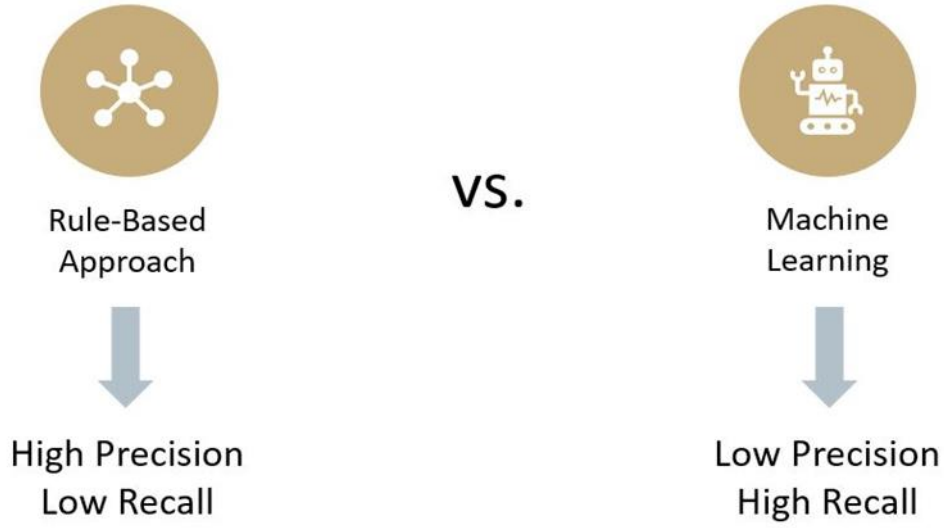

Figure I.5:Comparing between rule based and AI powered Chatbots [20].

I.6.3. Chatbots architecture

A chatbots work by leveraging a combination of technologies and techniques to understand user inputs, process them, and generate appropriate responses. Chatbot architecture refers to the underlying structure and components that enable a chatbot to function. It typically includes elements such as the user interface, natural language processing (NLP), dialog management, knowledge base, machine learning models, backend infrastructure, and analytics. These components work together to facilitate user interactions, understand queries, generate responses, and provide a seamless chatbot experience [15].

I.6.3.1. User Interface (UI)

The user interface is the front-end component of the chatbot that enables users to interact with the system. It can take various forms, including a website widget, messaging platforms (Facebook Messenger), or voice assistants (Amazon Alexa). The interface allows users to input their queries or commands and receive responses from the chatbot.

I.6.3.2. Natural Language Processing (NLP)

NLP is a critical component of chatbot architecture responsible for understanding and interpreting user input. It involves several subtasks such as text tokenization, named entity recognition, part-of-speech tagging, and sentiment analysis. NLP algorithms help the chatbot extract meaning from user messages, enabling it to generate appropriate responses.

I.6.3.3. Dialog Management

Dialog management handles the flow of conversation between the user and the chatbot. It ensures that the chatbot understands the context and maintains a coherent conversation. Dialog management systems use techniques like state tracking, intent recognition, and dialogue policies to determine the appropriate responses based on the current dialogue state.

I.6.3.4. Knowledge Base

A knowledge base is a repository of information that the chatbot can refer to when responding to user queries. It contains a collection of predefined responses, frequently asked questions, or relevant data from which the chatbot can retrieve information. Knowledge bases can be static or dynamic, with the latter being continuously updated based on user interactions and feedback.

I.6.3.5. Machine Learning Models

Machine learning models play a crucial role in enhancing chatbot capabilities. They enable the chatbot to learn from data and improve its performance over time. For example, machine learning models can be used for intent classification, entity extraction, response generation, or

even personalized recommendations. These models are trained on large datasets and can be updated regularly to adapt to changing user needs.

I.6.3.6. Backend Infrastructure

The backend infrastructure handles the processing and computation required to power the chatbot. It may include servers, databases, APIs, and other technologies necessary for handling user requests and generating responses. The backend infrastructure also ensures scalability, reliability, and security of the chatbot system.

I.6.3.7. Analytics and Monitoring

Analytics and monitoring tools provide insights into chatbot performance and user interactions. They help track metrics like user satisfaction, response accuracy, conversation length, and frequently asked questions. These insights can be used to fine-tune the chatbot's behavior, identify areas for improvement, and measure the effectiveness of the system [16].

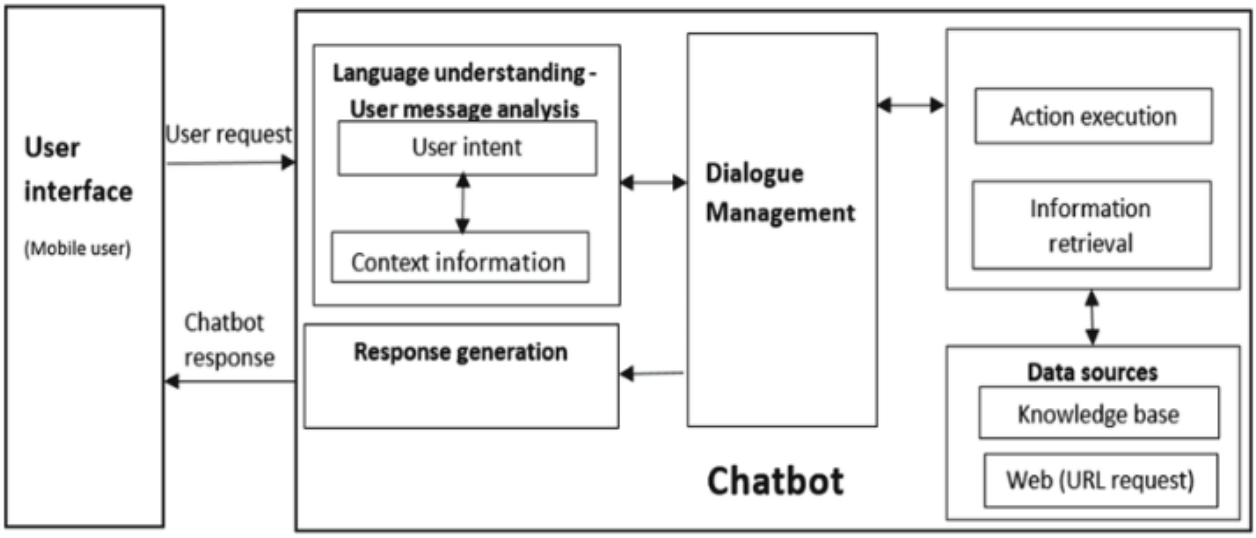

Figure I.6: Chatbots architecture [16].

I.6.4. How a chatbot works

Here's a step-by-step breakdown of how the architecture of an AI chatbot typically operates:

- **Step 1: User Interaction**

- The user interacts with the chatbot through a user interface, which could be a web interface, messaging app, voice assistant, or any other platform that facilitates communication with the chatbot.

- **Step 2: Natural Language Understanding (NLU)**

- The user's input is passed to the NLU component.

- NLU processes the user's message using techniques like tokenization, part-of-speech tagging, and named entity recognition.

- NLU aims to understand the user's intent (e.g., book a ticket, get train schedule) and extract relevant entities or information (e.g., station name, travel date).

- **Step 3: Dialogue Management**

- The dialogue management component manages the conversation flow and context.

- It maintains the state of the conversation, keeping track of previous user inputs and system responses.

- Based on the current conversation context, dialogue management determines the appropriate actions or responses.

- Dialogue management can be rule-based, where predefined rules govern the conversation, or machine learning-based, where a trained model makes decisions.

- **Step 4: Knowledge Base**

- The chatbot accesses a knowledge base that contains relevant information or data.

- The knowledge base can include structured data, FAQs, product details, or any other domain-specific information.

- The chatbot queries the knowledge base to retrieve the necessary information to respond accurately to the user's query.

- **Step 5: Backend Integration**

- If the chatbot needs to perform actions or retrieve data from external systems, it integrates with back-end services or APIs.

- For example, in the case of a railway ticket booking chatbot, integration with the railway ticketing system allows the chatbot to book tickets or access train schedules.

- **Step 6: Response Generation**

- Once the chatbot understands the user's intent and gathers any required information, it generates an appropriate response.

- The response can be pre-defined based on specific rules or dynamically generated using natural language generation (NLG) techniques.

- NLG may involve template-based responses or more advanced methods like sequence-to-sequence models that generate human-like text.

- **Step 7: Presentation and Delivery**

- The chatbot presents the generated response to the user through the user interface.

- The response can be displayed as text, voice output, or any other suitable format based on the chatbot's interface.

These steps represent a high-level overview of how an AI chatbot architecture typically functions. However, it's important to note that the actual implementation and details can vary depending on the specific chatbot's requirements, the technology stack used, and the desired functionalities [17].

I.6.5. Advantages and limitations of chatbots

Chatbots have become increasingly prevalent in today's digital landscape, offering a range of advantages but at the same time, like every modern technology, it has a number of limitations that makes it not fully effective. At this section, here are some of the main advantages and limitations of these chatbots:

I.6.5.1. Advantages

- **24/7 Availability**

Chatbots can provide round-the-clock customer support, allowing businesses to offer assistance and resolve queries at any time, regardless of their working hours.

- **Scalability**

Chatbots can handle multiple conversations simultaneously, which makes them highly scalable. They can handle a large volume of inquiries without experiencing delays or impacting the quality of service.

- **Cost-Effective**

Implementing a chatbot can be more cost-effective than hiring and training a large team of customer support representatives. Once developed and deployed, chatbots can handle customer inquiries without incurring additional costs.

- **Personalization**

Advanced chatbots can use machine learning algorithms to gather and analyze customer data, allowing them to personalize interactions. By understanding customer preferences and behavior, chatbots can offer tailored recommendations and suggestions.

- **Improved Customer Experience**

Chatbots can offer quick and consistent responses to customer queries, reducing waiting times and providing instant assistance. They can guide customers through various processes, ensuring a smooth and seamless experience.

- **Data Collection and Analysis**

Chatbots can gather valuable customer data during conversations, such as preferences, feedback, and common issues. This data can be analyzed to identify patterns, improve products or services, and enhance the overall customer experience [18].

I.6.5.2. Limitations

- **Lack of Human Touch**

Chatbots, particularly those based on rule-based systems, may struggle to provide the warmth and empathy of human interaction. They can sometimes come across as impersonal and fail to understand complex emotions or nuanced requests.

- **Limited Contextual Understanding**

Chatbots often struggle to understand context and can misinterpret queries, leading to inaccurate responses. They rely on predefined rules or keyword matching, which can result in misunderstandings or incomplete answers.

- **Handling Complex Queries**

While chatbots excel at handling routine and repetitive tasks, they can struggle with complex or unique inquiries that fall outside their programmed capabilities. In such cases, they may need to escalate the conversation to a human agent.

- **Language Limitations**

Chatbots can be language-dependent, and building multilingual chatbots can be challenging. They may not accurately understand and respond to slang, colloquial language, or cultural nuances, limiting their effectiveness in diverse markets.

- **Initial Development and Maintenance**

Developing a chatbot requires time, resources, and expertise. It involves designing conversational flows, integrating with backend systems, and continuous monitoring and updates to improve performance and accuracy.

- **Security and Privacy Concerns**

Chatbots can interact with sensitive customer information, such as personal details or account credentials. Ensuring robust security measures and compliance with privacy regulations is crucial to protect user data from unauthorized access or breaches [19].

I.7. Conclusion

In This first chapter we talked about Artificial intelligence in general, its subsections called machine learning a deep learning, after that we took an overview on Chatbots, their types, how it works and its architecture. Knowing these basic information's mentioned before, will be like an entry to our next chapter where we will go deeper and talk about the methods used in building and implementing Chatbots.

.

Chapter II:

Methods used

II.1. Introduction

In this chapter, we are going to explain the ML techniques used on building different types of chatbots, by giving some exact and deep definitions of these techniques, we can say that we are going to track the chatbot's Background and what happens at the same time step by step.

II.2. What is a machine learning Chatbot

Behind the scenes of any conversational AI-powered agent lies a powerful combination of Neural Networks, Machine Learning, and Natural Language Processing (NLP). This chapter introduces the background of building chatbots using these technologies, exploring the key components, techniques, and considerations involved in their development.

At the heart of chatbot development lies the use of Neural Networks, which are computational models inspired by the structure and functioning of the human brain. Neural Networks, particularly recurrent neural networks (RNNs), have proven to be highly effective in capturing the sequential nature of conversations, making them suitable for chatbot applications. These networks excel in learning patterns, context, and generating meaningful responses based on input data [20].

II.3. Neural networks

II.3.1. Definition

A neural network (NN) is a reflection of the human brain's behavior. It allows computer programs to recognize patterns and solve problems in the fields of machine learning, deep learning, and artificial intelligence. These systems are known as artificial neural networks (ANNs) or simulated neural networks (SNNs). Google's search algorithm is a fine example [21].

In another way**, a Neural Network is essentially a network of mathematical equations**. It takes one or more input variables, and by going through a network of equations, results in one or more output variables. You can also say that a neural network takes in a vector of inputs and returns a vector of outputs, but I won't get into matrices in this article [22].

Neural networks are subtypes of machine learning and form the core part of deep learning algorithms. Their structure is designed to resemble the human brain, which makes biological neurons signal to one another. ANNs contain node layers that comprise input, one or more hidden layers, and an output layer [23].

II.3.2. The Mechanics of a Basic Neural Network

Neural networks are easy to understand but it gets hardrer when we try to get too deep into its mechanics, but it's worthwhile to show you what the structure of a basic neural network looks like[38].

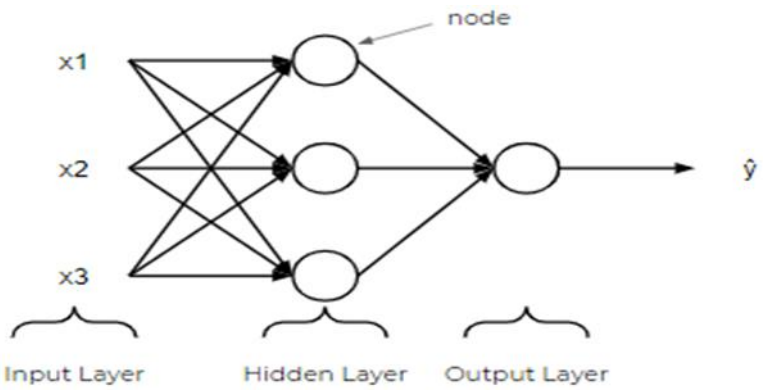

Figure II.6: Neural network architecture with the three main layers [W2].

In a neural network, there's an **input layer**, one or more **hidden layers**, and an **output layer**. The input layer consists of one or more feature variables (or input variables or independent variables) noted as x1, x2, … The hidden layer consists of one or more hidden **nodes** or hidden units. A node is simply one of the circles in the diagram above. Similarly, the output variable consists of one or more output units[W3].

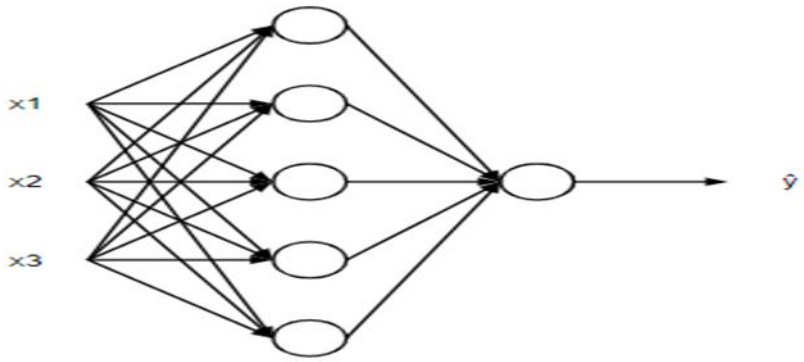

Figure II.7: An NN layer with many nodes [W3].

Like we said earlier, a neural network is nothing more than a network of equations. Each node in a neural network is composed of two functions, a linear function and an activation function. This is where things can get a little confusing, but for now, think of the linear function as some

line of best fit. Also, think of the activation function like a light switch, which results in a number between 1 or 0.

What happens is that the input features (x) are fed into the linear function of each node, resulting in a value, z. Then, the value z is fed into the activation function, which determines if the light switch turns on or not (between 0 and 1) [24].

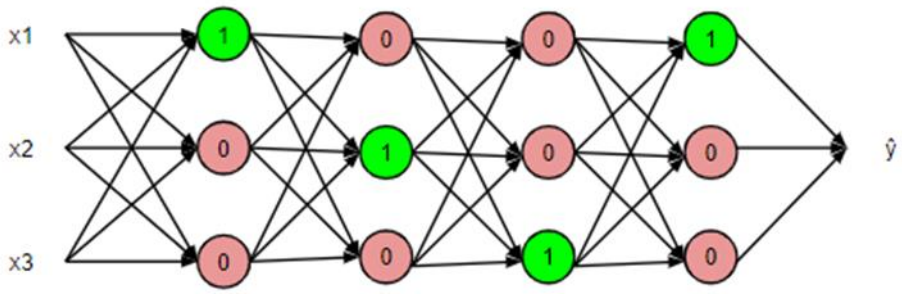

Figure II.8 : A neural network presses in a metaethical overview [24].

So, each ultimately determines which nodes in the following layer get activated, until it reaches an output. Conceptually, that is the essence of a neural network.

II.3.3. Types of neural networks

Neural network models are of different types and are based on their purpose, there are three main types we can classify it in, it depends on the purpose we need it to [W4].

The most common types of neural networks are:

II.3.3.1. Single-layer perceptron

The perceptron created by Frank Rosenblatt is the first neural network. It contains a single neuron and it is very simple in structure.

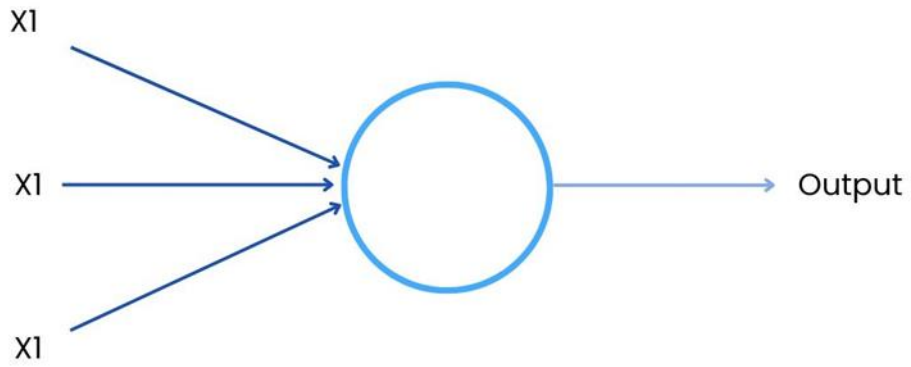

Figure II.9: A single layer neural network [W4].

II.3.3.2. Multilayer perceptrons (MLPs)

These form the base for natural language processing (NLP). They contain an input layer, a hidden layer, and an output layer. It is important to know that MLPs contain sigmoid neurons and not perceptrons because most real-world problems are non-linear. Data is fed into these modules to train them.

- **Artificial neural networks (ANNs)**

Most folks consider Artificial Neural Networks (ANN) with Neural Networks (NN). However, both aren't one and the same. Instead, an ANN is a type of Neural Network.

ANN is essentially a feed-forward network because of the way information travels through it from one layer to another without touching a node twice [W4].

This type of neural system patterned around how neurons work in our brain recognizes patterns in raw data, helping solve complex processes. Another similarity with the human brain: the ANN improves with every new input it gets.

Now to dive into the composition of this Neural Network. The ANN is based on three or more interconnected layers of nodes again, similar to the brain. All layers are responsible for inputting, processing, and outputting data to the deeper layers. Such an interlayered system is what helps this Neural Network understand and learn complex things [25].

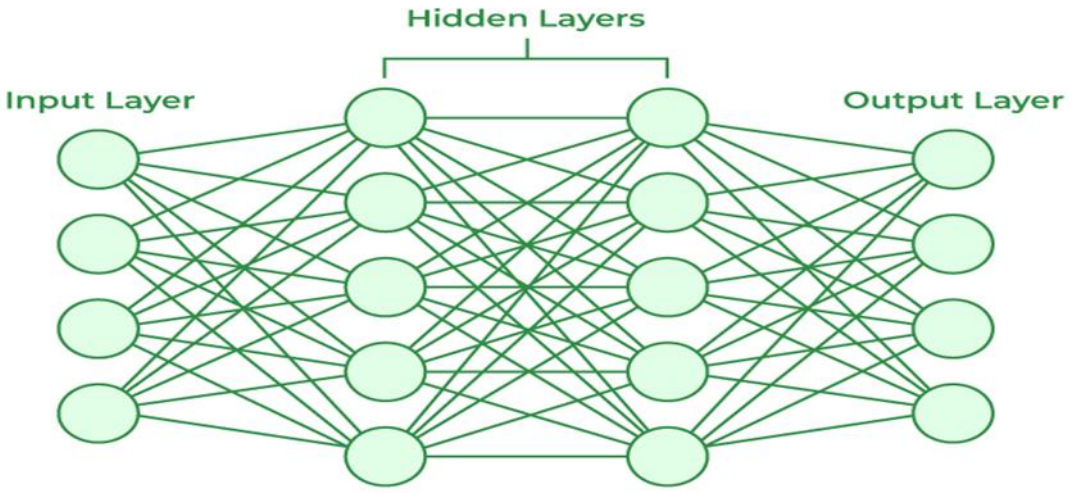

Figure II.10: An ANN model with more than 3 layers [26]**.**

- **Convolutional neural networks (CNNs)**

Convolution Neural Networks are mainly credited for their role in image and video recognition, recommendation systems, and image analysis and classification. Prior to it, manual labor was involved in identifying objects in images. CNN, however, has helped scale the process by using linear algebra principles to identify patterns in images.

So how do CNNs work? CNNs are based on three main layers. These are :

- ✓ Convolutional layer
- ✓ Pooling layer
- ✓ Fully-connected layer

With each layer, the CNN's complexity in understanding the image increases. This means the first layers focus on interpreting simple features in an image such as its edges and colors. As the image processes through layers, the network is able to recognize complex features such as object shapes. Finally, the deepest layer is able to identify the target object [26].

What should you use CNNs for?

This type of Neural Network is used for Computer Vision. Essentially, Computer Vision is an AI field that's responsible for identifying meaningful information from visual inputs, images, and videos. For example, identifying dogs from a pile of images.

Such CNN-powered Computer Vision has several applications including:

- ✓ Recommend products that will likely complement an existing product or wardrobe.
- ✓ Provide suggestions for tagging people in a photograph as social media platform do.
- ✓ Identify items as Computer Vision currently does for helping doctors better recognize cancerous tumors.

Using the CNNs show one main advantage, and it is that It automatically detects features without human supervision**.** This gives it an edge over its predecessors [27].

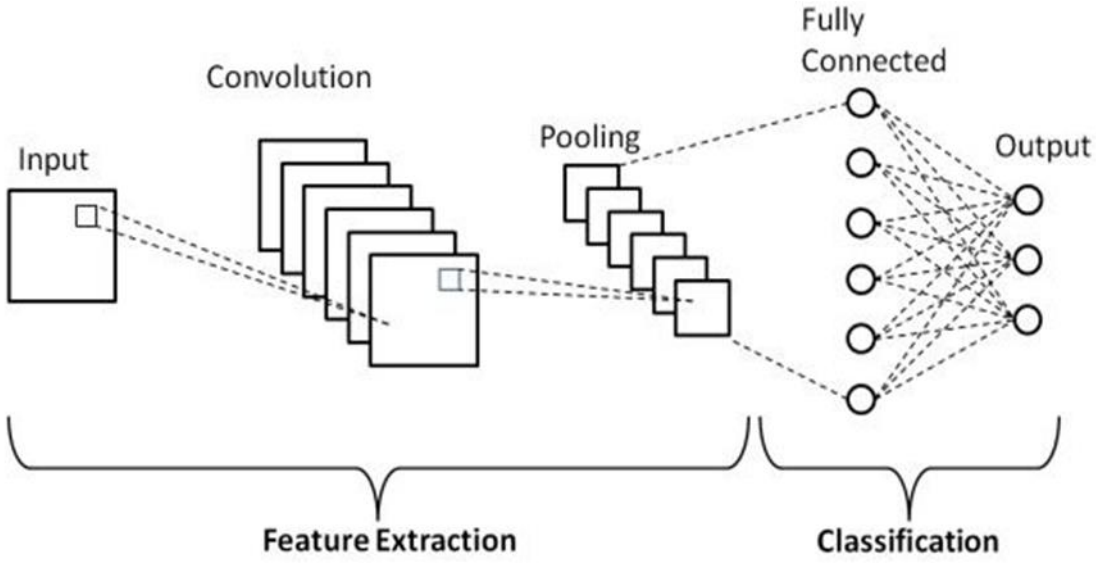

Figure II.11: CNN model architecture [27].

- ➢ **Recurrent neural networks (RNNs)**

A recurrent neural network (RNN) is a type of advanced artificial neural network (ANN) that involves directed cycles in memory. One aspect of recurrent neural networks is the ability to build on earlier types of networks with fixed-size input vectors and output vectors.
The use of recurrent neural networks are often related to deep learning and the use of sequences to evolve models that simulate the neural activity in the human brain.

In terms of practical application, RNNs have been an active area of focus for many professionals for uses like image processing, language processing, and even models that add characters to text one at a time. By playing around with these text generation models, scientists have been able to produce samples that look a lot like different kinds of human writing – for example, modern investment op-eds, or classical Shakespeare plays. The RNN has been able to

generate text results that demonstrate the ability to learn English from scratch, or from very limited programming inputs.

Many examples of using RNNs produce text that is not grammatically correct. The idea is that a large number of these experiments and systems need additional supports to really become useful – but they do demonstrate amazing artificial intelligence power to model the human generation of language [28].

II.4. Natural language processing (NLP)

II.4.1. Definition

NLP, an abbreviation of Natural language processing is nothing but an engine for the chatbot to understand the user's intent in the message and fetch the most appropriate response from its database. Regardless of which language a computer is learning, NLP understands the syntax, semantics, discourse, and purpose of the message to engage in a human-like conversation [29].

It focuses on how we can program computers to process large amounts of natural language data in a way that is productive and efficient, taking certain tasks off the hands of humans and allowing for a machine to handle certain processes [W5]. NLP takes it into account the following:

- ✓ Named Entity Recognition (NER) or 'entity identification' locates and classifies named entity mentions in unstructured text into predefined categories.
- ✓ Part-of-Speech (POS) tagging or 'tokenization' which is reading a text in some language and assign parts of speech to each word (and other tokens), such as noun, verb, adjective, etc.
- ✓ Text categorization labels natural language texts with relevant categories from a predefined set.
- ✓ Syntactic parsing that analyses a string of symbols to adhere to the rules of a formal grammar [30].

II.4.2. Natural language processing Components

There are two components of an NLP system: Natural Language Understanding (NLU) and Natural Language Generation (NLG).

When you input a text into an NLP engine, the meaning or context of the user is deciphered by the NLU construct and the response is generated by NLG. The following equation best explains the relationship between NLP, NLU, and NLG [30]:

NLP = NLU + NLG

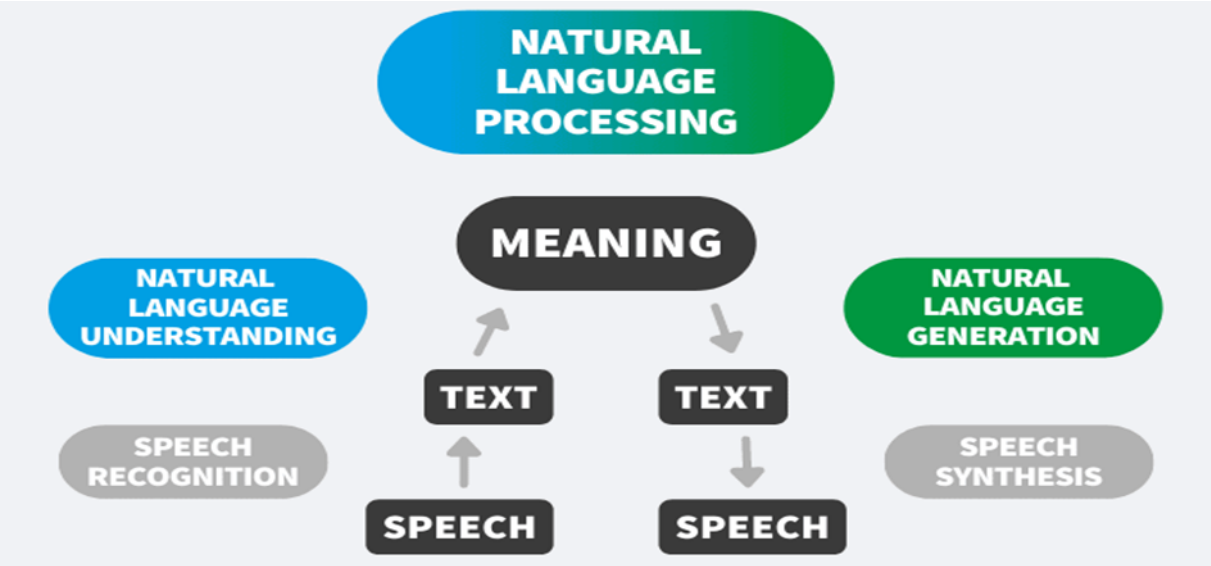

Figure II.12 : NLP components [30].

Process flow

User Text -> Chatbot -> NLU -> Meaning deciphered -> NLG -> Response Generated

II.4.2.1. Natural Language Understanding (NLU)

NLU is understanding the meaning of the user's input. Primarily focused on machine reading comprehension, NLU gets the chatbot to comprehend what a body of text means. NLU is nothing but an understanding of the text given and classifying it into proper intents.

NLU can be applied to a lot of processes. From categorizing text, gathering news and archiving individual pieces of text to analyzing content, it's all possible with NLU [31].

Real-world examples of NLU include small tasks like issuing short commands based on text comprehension to some small degree like redirecting an email.

to the right receiver based on basic syntax and decently sized lexicon.

NLU involves :

- **Natural Language Inference (NLI) and paraphrasing**: it determines whether a statement is true (entailment), false (contradiction), or undetermined (neutral). This is done by setting a 'premise' for the system in the form of a training database.

- **Dialogue agent**: these managers keep track of the current state of conversations.
- **Semantic parsing**: converts natural language utterance to a logical form that is understandable by machine
- **Question answering**: it automatically answers in a natural language.
- **Sentiment analysis**: uses text analysis, computational linguistics, and biometrics to systematically identify, extract, and quantify subjective information.
- **Summarization**: shortening the message and emphasizing the major points (intent/entity) [32].

II.4.2.2. Natural Language Generation (NLG)

NLG is a software that produces understandable texts in human languages. NLG techniques provide ideas on how to build symbiotic systems that can take advantage of the knowledge and capabilities of both humans and machines.

The input can be any non-linguistic representation of information and the output can be any text embodied as a part of a document, report, explanation, or any other help message within a speech stream. The knowledge source that goes to the NLG can be any communicative database [33].

The NLG process is carried out in 7 different steps of document planning, micro-planning, and linguistic realization.

- **Content determination**: This enables a Generative Conversational AI to decide what to respond - the determination process itself is based on lengthy training data that clearly outlines possible intent and entity sets.
- **Discourse planning**: This is required to structure the texts based on conceptual grouping and rhetorical relationships.
- **Sentence aggregation**: This is done to produce large, complex sentences by combining messages in a sentence plan.
- **Lexicalisation**: Depending upon the developer's intent, this is done to identify domain-centric jargons.
- **Referring expression generation**: This covers the usage of reference - nouns, pronouns, definite description, etc.
- **Syntactic and morphological realization**: The rudimentary concepts of grammar like morphology and syntax are fed into the system at this stage.

- **Orthographic realization**: Last for language generation is the correct use of punctuation and casing and typographic requirements like font size [34].

II.4.3. Conversations with a meaning

Deploying a rule-based chatbot can only help in handling a portion of the user traffic and answering FAQs. NLP (i.e. NLU and NLG) on the other hand, can provide an understanding of what the customers "say". Without NLP, a chatbot cannot meaningfully differentiate between responses like "Hello" and "Goodbye". They will both be nothing more than text-based user inputs.

NLP-driven intelligent chatbots can, therefore, improve the customer experience significantly. Customers all around the world want to engage with brands in a bi-directional communication where they not only receive information but can also convey their wishes and requirements. Given its contextual reliance, an intelligent chatbot can imitate that level of understanding and analysis well. Within semi-restricted contexts, it can assess the user's objective and accomplish the required tasks in the form of a self-service interaction. Such a chatbot builds a persona of customer support with immediate responses, zero downtime, round the clock and consistent execution, and multilingual responses [29].

II.4.4. With Chatbots, Possibilities are Endless

NLP based chatbots not only increase growth and profitability but also elevate customer experience to the next level all the while smoothening the business processes. Together with Artificial Intelligence/ Cognitive Computing, NLP makes it possible to easily comprehend the meaning of words in the context in which they appear, considering also abbreviations, acronyms, slang, etc. This offers a great opportunity for companies to capture strategic information such as preferences, opinions, buying habits, or sentiments. Companies can utilize this information to identify trends, detect operational risks, and derive actionable insights [35]. NLP, NLU, and NLG can prove to be fruitful investments as they provide:

- **Immediate assistance**: Like a normal chatbot, they can engage in conversations 24*7 with responses on the fingertips and minimal lag time
- **Efficient service**: Precise understanding of user intentions and mapping it onto relevant responses.
- **More Human-like engagement**: Personalized, one-to-one experience, in a conversational style is delivered creating a better, humane impression.

- **Cost and time saving**: Reduce workload from human operators even for complex tasks and save time.

Additionally, NLP can be coupled with budding technologies like phenomenon modelling, intelligent decisions, reasoning, and autonomy to artificial general intelligence (AGI) to weave an even more personalized and unique customer engagement. It can identify spelling and grammatical errors and interpret the intended message despite the mistakes. This can have a profound impact on a chatbot's ability to carry on a successful conversation with a user [36].

II.5. Conclusion

In this chapter, we could define in a brief way the main Techniques that building an AI-powered chatbot requires. Starting with machine learning and the Neural networks system, how it works and its types. After that we defined the Natural language processing and its subsections NLU and NLG.

These techniques are essential in our "Building a chatbot" project, so it's very important to have this information well understood, so that we can build a strong AI-POWERED AGENT, and this is what we will achieve at the ending of the next chapter called "**Building and implementation".**

Chapter III:

Building and implementation

III.1. Introduction

In this final chapter, we are going to build two chatbots: API powered chatbot, and a customer service chatbot) step by step and by using several tools, and after the implementation we will discuss our results a compare them to each other and see if we can achieve our main goal which is improving our customer services.

III.2. Tools

As we all know, building any type of chatbot requires a good knowledge of some programming languages, some specific platforms and many other several tools. Here are the main tools and platforms we used to achieve our Chatbots building and implementing:

III.2.1. Applications and platforms

III.2.1.1. Visual Studio Code (VS Code)

Visual Studio Code (VS Code) is a lightweight and highly popular source code editor developed by Microsoft. It is known for its versatility, extensibility, and support for a wide range of programming languages. VS Code provides a rich set of features such as code completion, debugging, version control integration, and customizable user interface, making it a preferred choice for many developers.

When it comes to Python development, Visual Studio Code provides excellent support. It offers features specifically tailored for Python development, such as syntax highlighting, code formatting, code navigation, linting, and debugging capabilities. Overall, VS Code is a popular choice among Python developers due to its efficiency, ease of use, and robust Python-specific features [W10].

Figure III.13: Visual Studio Code [W10]

III.2.1.2. API Keys by Openai

An application programming interface (**API**) is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces

of software. A document or standard that describes how to build or use such a connection or interface is called an API specification. A computer system that meets this standard is said to implement or expose an API. The term API may refer either to the specification or to the implementation [W11].

API keys are a way to authenticate and authorize access to OpenAI's API services. They are used to identify the developer or application that is making the API request, and to ensure that the API is only being used in accordance with the terms of service. API keys can be used to track usage and bill for API usage, as well as to provide additional security and access control. For example, API keys can be used to limit the number of requests per minute, or to restrict access to certain features or data to use OpenAI's API services, you will need to create an account and generate an API key. You will then need to include the API key in the headers of your API requests.

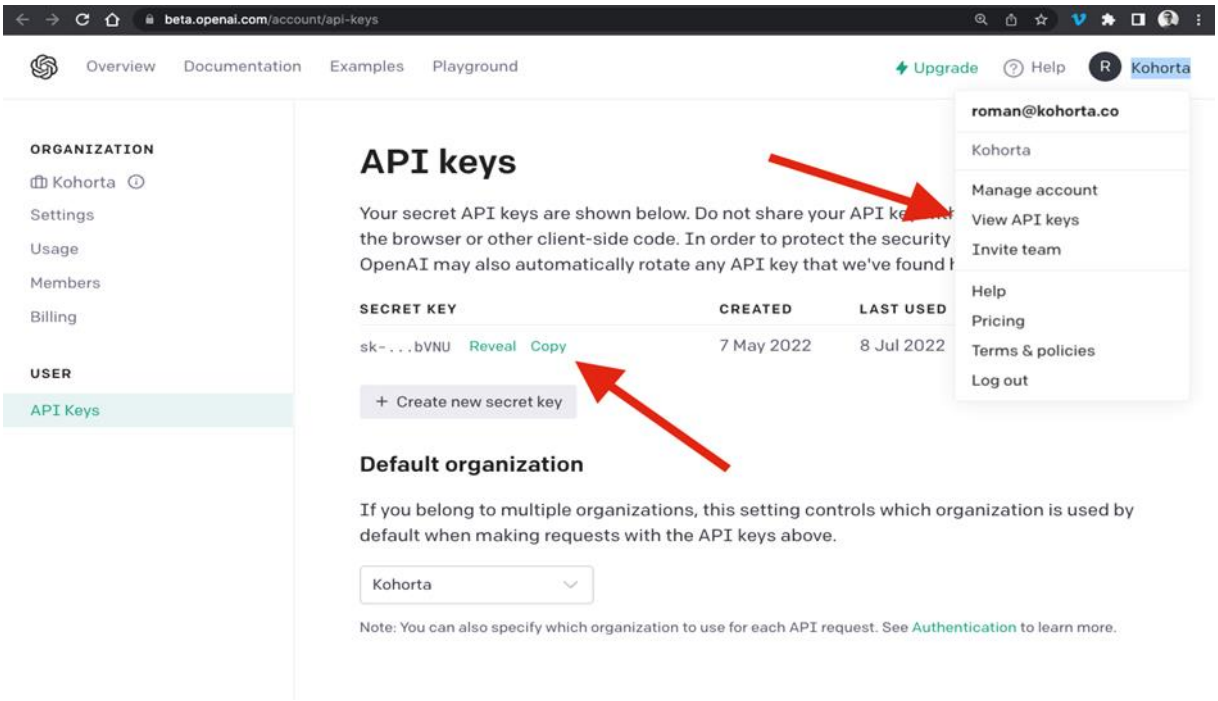

Figure III.14:OpenAI API key generator [W11].

III.2.1.3. Google Colaboratory

Google Collaboratory, commonly referred to as Google Colab or Colab, is a cloud-based platform that provides users with an easy-to-use environment for writing, running, and sharing Python code. It is based on the popular Jupyter Notebook interface and runs entirely in the cloud, which means that users do not need to install any software on their local machines to use it.

One of the key features of Colab is that it provides free access to powerful hardware, including GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units), which can greatly accelerate computationally intensive tasks such as machine learning and deep learning. This makes Colab a popular choice among data scientists, researchers, and machine learning practitioners who need to perform complex computations on large datasets.

Colab also provides a variety of pre-installed libraries and tools, including popular data science and machine learning packages such as NumPy, Pandas, scikit-learn, TensorFlow, and PyTorch. This makes it easy for users to get started with their projects without having to worry about installing dependencies [W8].

Figure III.15: Google colaboratory [W8].

III.2.1.4. ChatBot.com

ChatBot.com is one of the best platforms for online businesses to build AI bots without having a lot of knowledge in coding programs. Many users highly recommend the application, achieving the best rating on comparison sites. The application is intuitive, easy to use, and does not require technical knowledge.

With ChatBot.com, you can use the framework and templates to create custom chatbots that can handle various tasks, from answering frequently asked questions in customer service to sending personalized recommendations and processing orders and bookings [W9].

The solution provides a drag-and-drop interface to build conversational bots without coding. A customizable chat widget lets you implement an AI bot on any website.

Adding to that, ChatBot platform provides you several integrations with other messaging applications, websites and more.

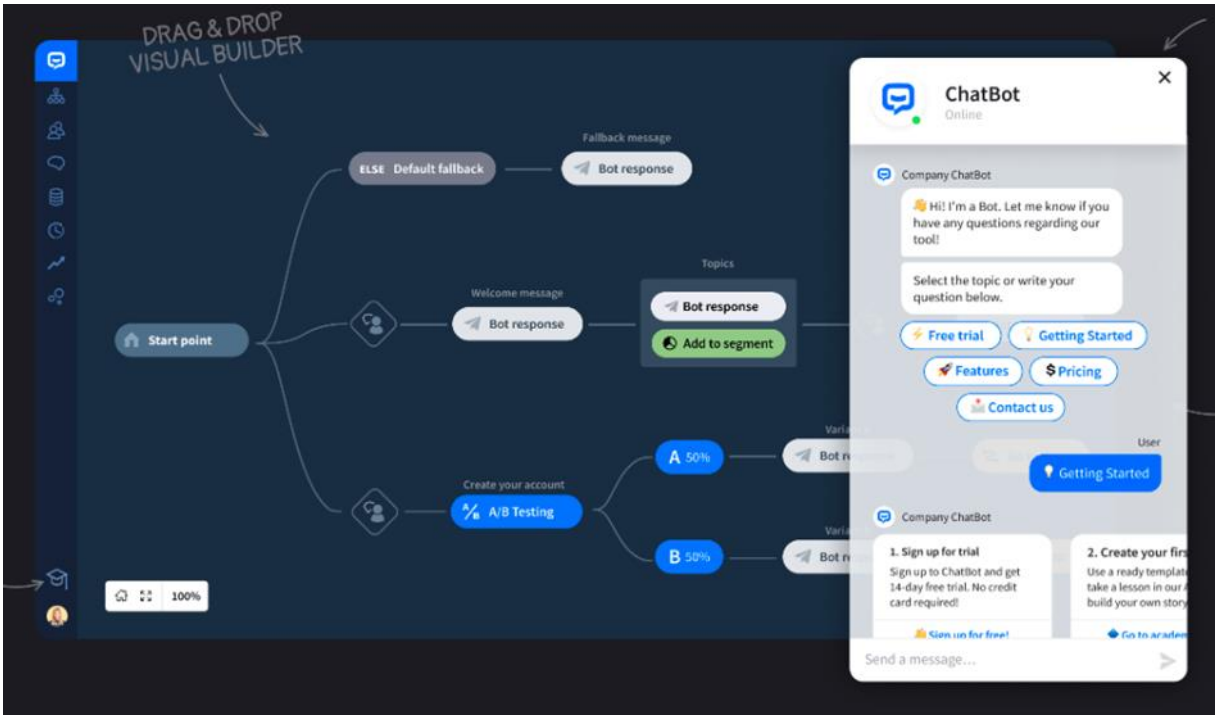

Figure III.16: ChatBot Platform [W9].

III.2.2. Programming language and libraries

III.2.2.1. Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today [W6].

Stack Overflow's 2022 Developer Survey revealed that Python is the fourth most popular programming language, with respondents saying that they use Python almost 50 percent of the time in their development work. Survey results also showed that Python is tied with Rust as the most-wanted technology, with 18% percent of developers who aren't using it already saying that they are interested in learning Python [W7].

Figure III.17: Python [W7].

III.2.2.2. JSON

JSON (JavaScript Object Notation) is an open-standard, text-based data format or interchange for semi-structured data. Semi-structured data is machine data that originates from a wide variety of sources and devices, including mobile phones, web browsers, servers, or IoT devices. This data, which is collected as messages called events, can then be organized into batches, which are fed to a data platform via a data pipeline.

JSON can be used in a multitude of applications, but it has become more common as a format for data transmission between servers and web applications or web-connected devices. This is partially because those applications can often only receive data as text, and JSON is text based. Originating from JavaScript, most contemporary programming languages now have the ability be able to parse and generate JSON data [W12].

Figure III.18: JSON [W12].

III.2.2.3. NLTK library

The Natural Language Toolkit (abbreviated as NLTK) is a collection of libraries designed to make it easier to process and work with human language data, so think something along the lines of corpora analysis and text processing. It was developed by Steven Bird and Edward Loper while they worked at the University of Pennsylvania, my own alma mater, and as a result the library tends to be used more for research or thesis related work. The toolkit is maintained by a team around the world that contribute to different modules, though anyone is able to pitch in and contribute to the toolkit, leading to integration with many other outside programs like the Stanford tagger.

The NLTK comes built in with many libraries to help with common NLP processes. These include the sort of suspects you'd typically expect, like a tokenizer that takes in a string and returns a list of strings of whatever units you specify to chunk for, as well as modules for lemmatization, which group together all inflected forms of a word to help make analysis easier and more comprehensive. Furthermore, there's modules for classification and sentiment analysis to help analyze and better understand given speech data. You can play around with some of these modules through their demos here. Overall, the toolkit is essentially a Swiss army knife for helping breakdown and efficiently analyze a corpus of linguistic information [W11].

Figure III.19: Natural language toolkit [W11].

III.3. Building and implementing our chatbot

Building a chatbot consists many steps called the workflow, in the figure below, we can see that the first step is to identify business use cases, then defining tasks that the bot would perform, next we will use the power of NLP to identify the need of Customers, and create a machine learning model that could understand and generate responses, after we are done with the model we have to train it and it's going to take a while, so we have to test it during the training phase, and when achieving almost 80% of valid results, we will be able to deploy our chatbot and make it accessible to our customers.

To start the designing process, several questions related to the use of the bot and how it will interact with users need to be answered, some of these questions are:

- ✓ What does the chatbot need to do?
- ✓ What business goals will the chatbot be made to achieve?
- ✓ How satisfying does the chatbot perform in front of customers?

All these questions and more need to be answered before starting to build a chatbot, but the main question must be:

- ✓ Does the chatbot have to be able to handle long conversations or short conversations?

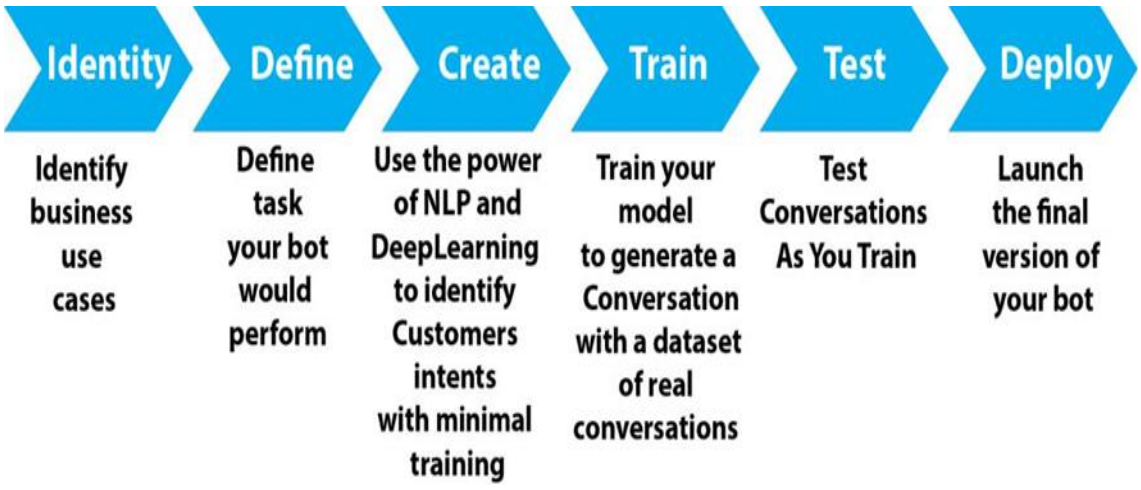

Figure III.20: Process of building a chatbot

III.3.1. API chatbot

III.3.1.1. Conception

As said before, we ae going to implement two types of chatbots, the first one is an easy model to build called ‘AnisGPT 1.0’ o build your own AI chatbot using the ChatGPT API keys. We have also implemented a Gradio interface so you can easily demo the AI model and share it with other people to interact with. We will use Google colab as an environment and Python as a programming language.

- ✓ The first step of building this chatbot, is to start by installing two main libraries called "openai" and "gradio" using the pip install command.

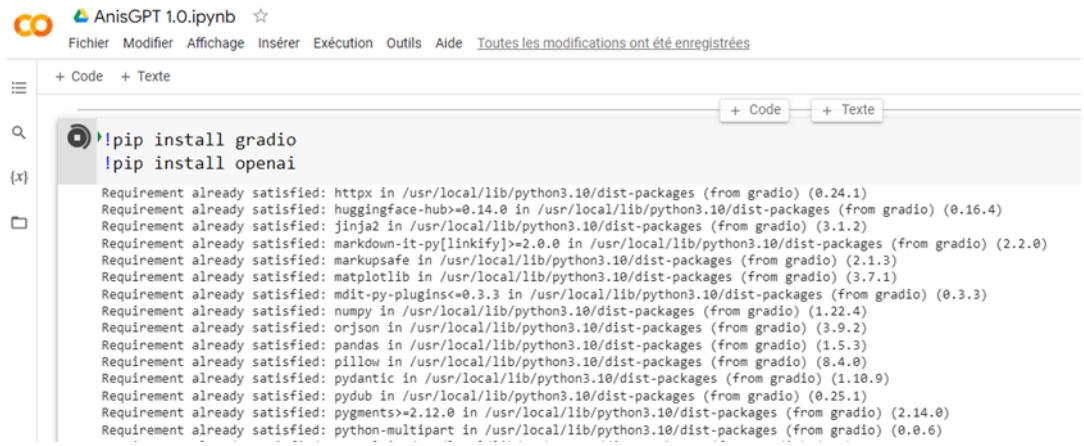

Figure III.21: Installing librairies

- ✓ After the libraries are installed, we have to get an API KEY generated by OPENAI site.

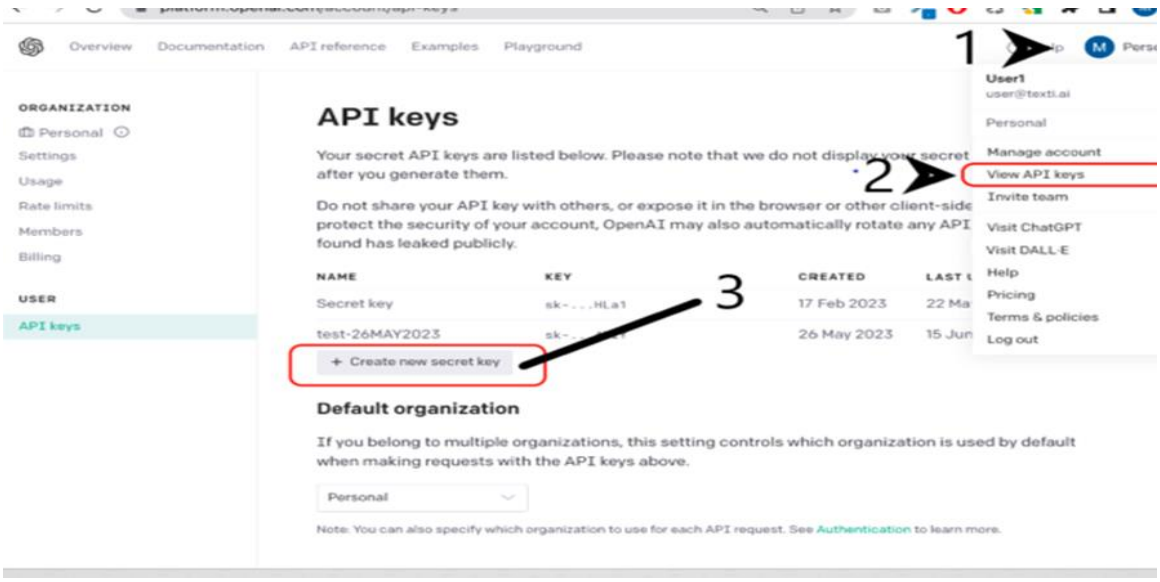

Figure III.22: Getting the API KEY

- ✓ Now we are ready to write our python program, just after the library's installation, we write the script showed in the image below, and integer our API KEY that we got before from openai platform.

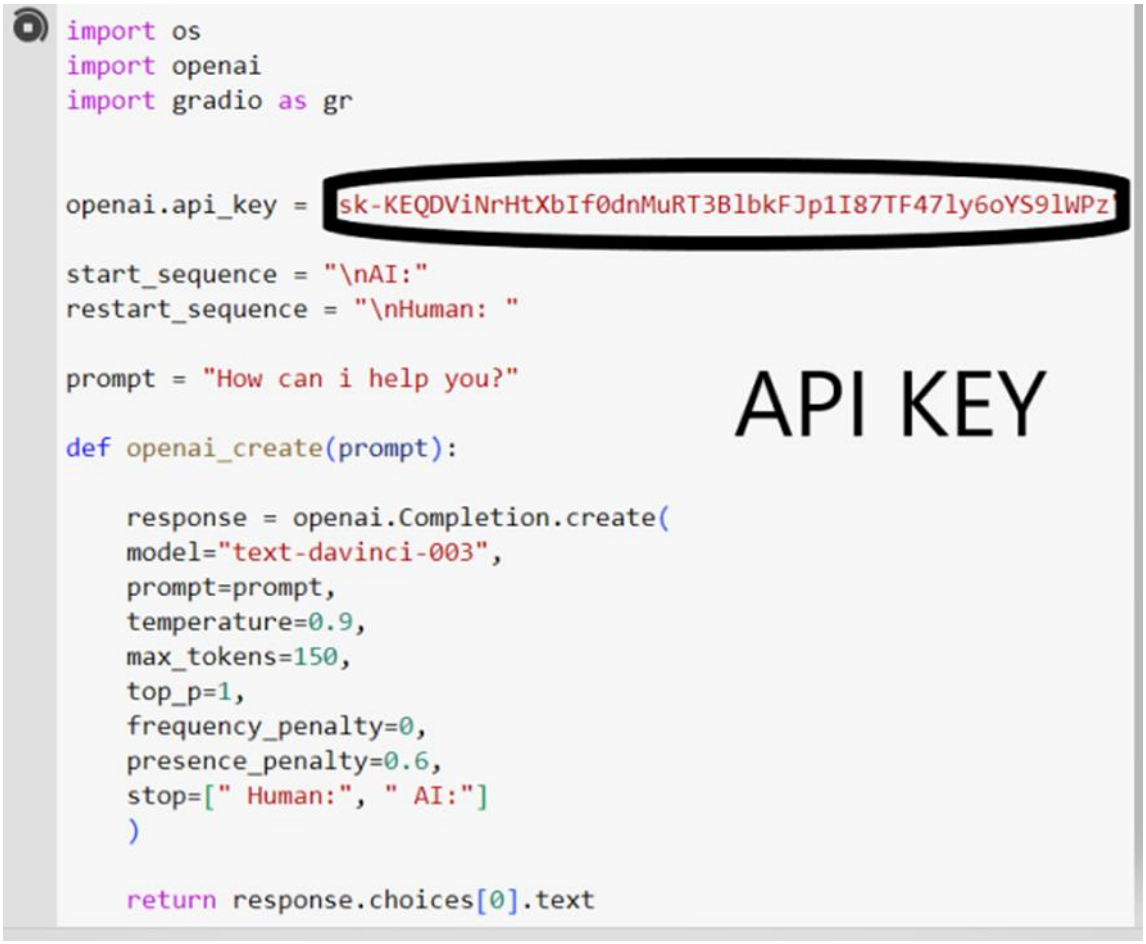

Figure III.23: AnisGPT 1.0 python program

- ✓ Finally, we set up the Gradio interface program section, and after the execution, the gradio interface will relate automatically the user into a hosting site that can be shared to be tested and valid for 72 hours before this link expires.

- ✓ If the link expired, you have to either change your API KEY bey generating a new one, or re-execute your python program and the it will automatically provide you with a new hosting site.

- ✓ A basic chat window will pop up after going to the Gradio's hosting site, it contains the main sections: The users input case, output for the models responses, and SEND button.

Our model named AnisGPT 1.0 is now available to be tested, and that's what we are going to see after this phase.

III.3.1.2. Test and results

After programming and training our model called AnisGPT 1.0, it's very important to notice that all API chatbots can't achieve the 100% of the responses validity, especially when knowing that OpenAi model named ChatGPT-3.5 turbo stopped the data updating in 2021, that means it can't provide you any response related to after 2021.

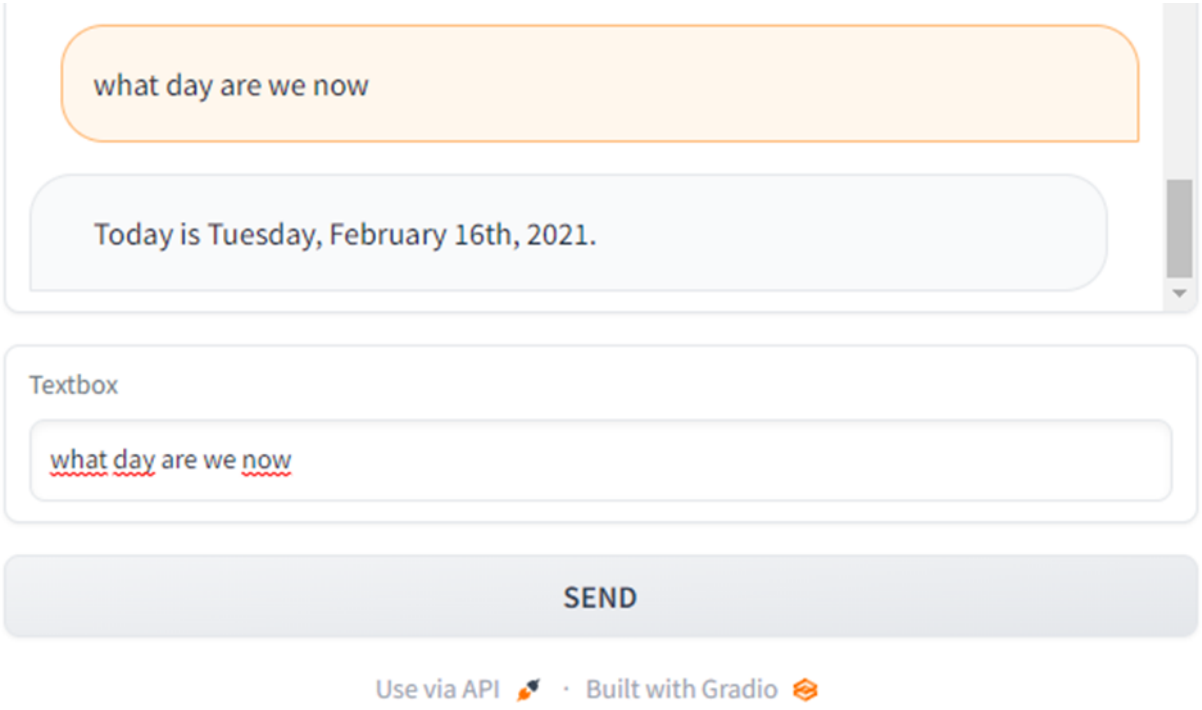

Figure III.24: API Chatbot date response.

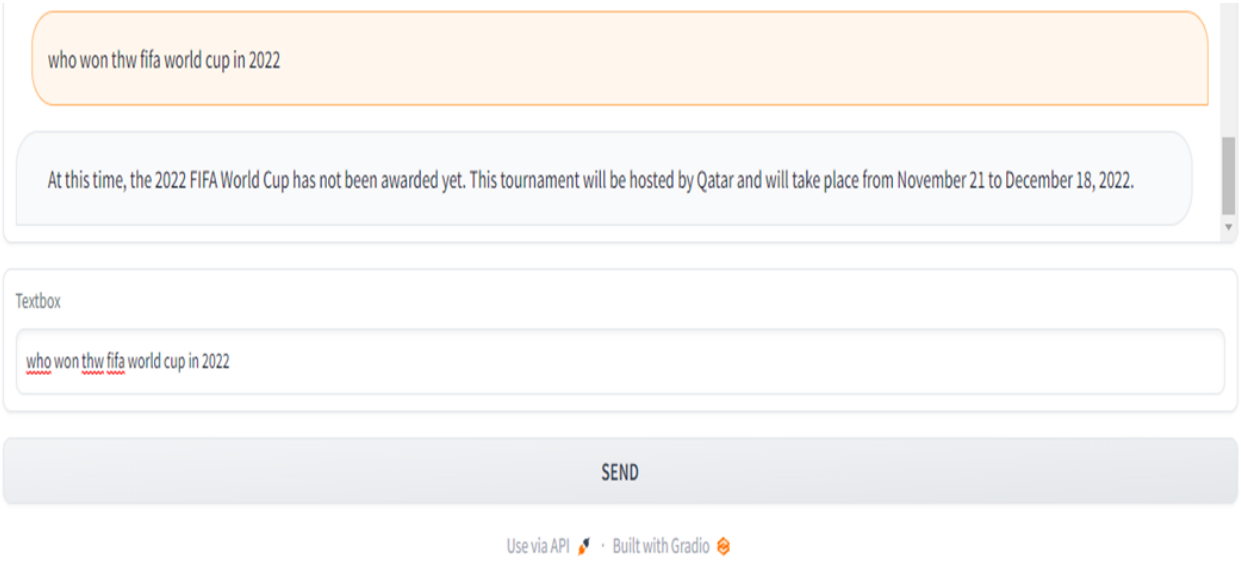

Figure III.25: API Chatbot second date response

But meanwhile, our chatbot could've generate several correct responses based on user's questions. Here are some common questions that my API chatbot generated correct answers for it:

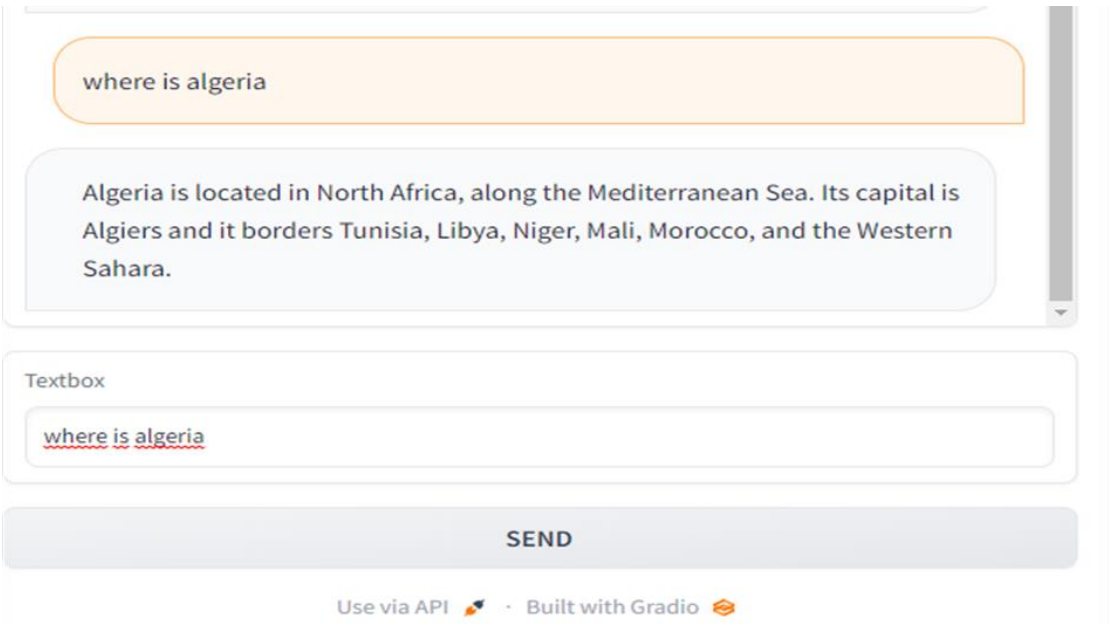

Figure III.26: First correct response from our model AnisGPT 1.0.

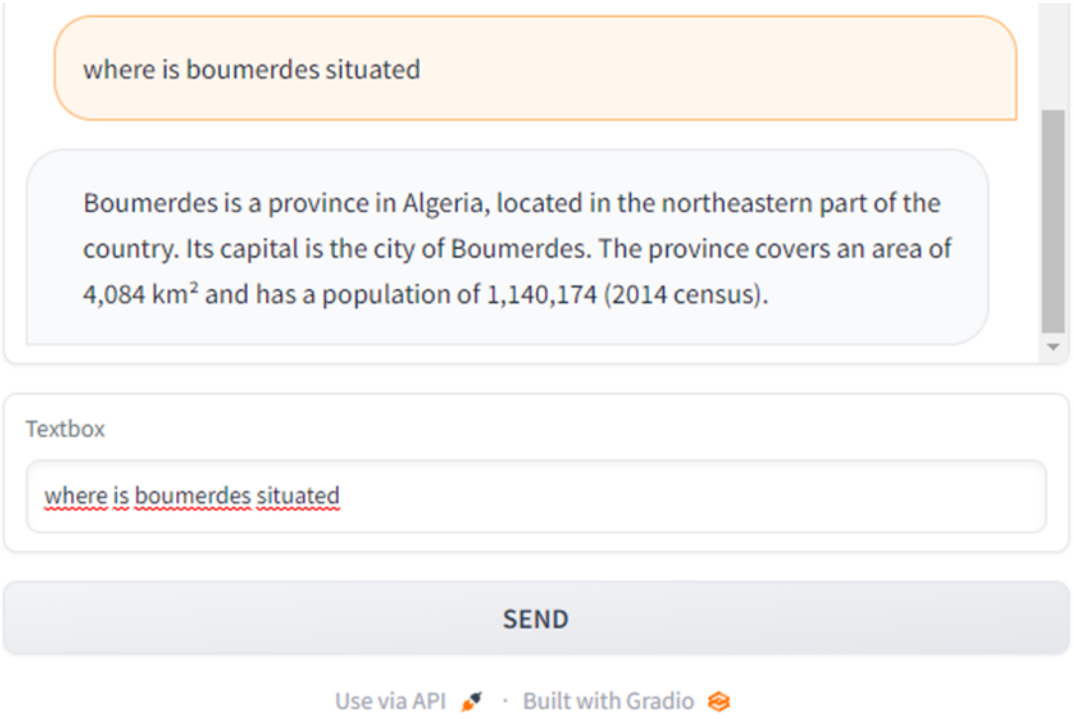

Figure III.27 : Second correct response by AnisGPT 1.0 model.

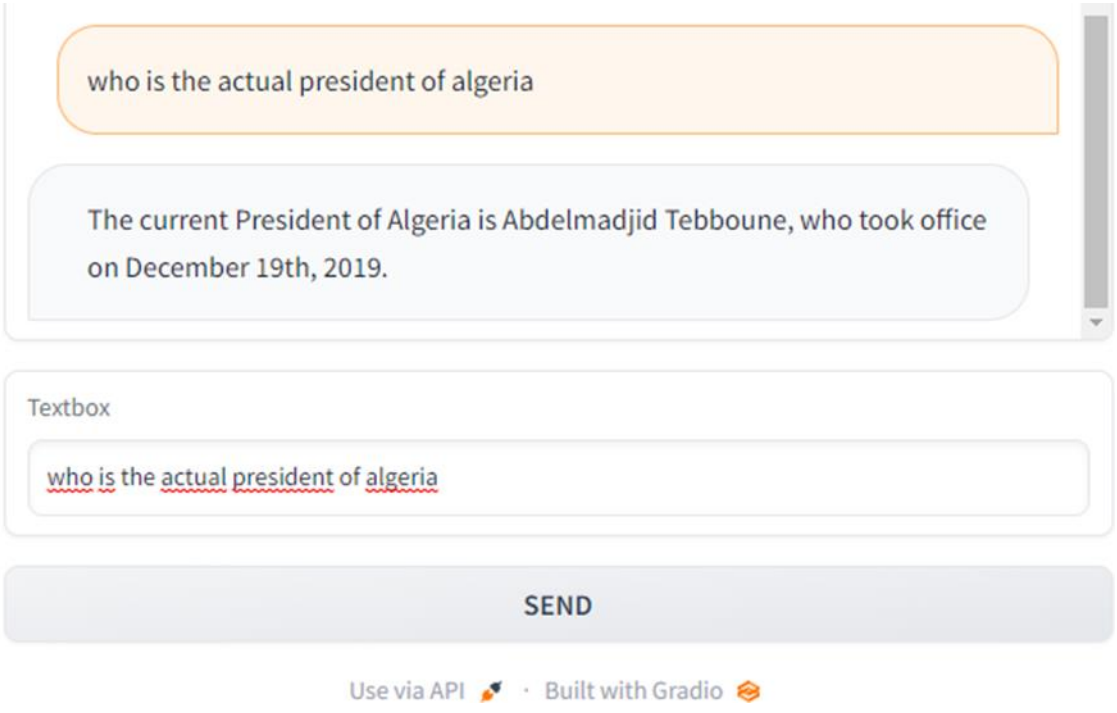

Figure III.28: Third correct response by AnisGPT 1.0 model.

III.3.1.3. Results discussion

Before the training phase, my chatbot showed me a low rate of accuracy, it answered 57 questions only of 100 asked, and that means that we should train our model more and more to improve its answers validity.

After the training phase of our model, we got the chance to ask it 100 exact questions, my it answered almost 85 of them, those 15 were empty outputs.

Above those 85 answered answers, were divided into two sections, wrong answers an.

So, we can say that our chatbot made an accuracy rate of 85% and the loss rate is only 15%.

To understand these results more, I represented it on the diagram below:

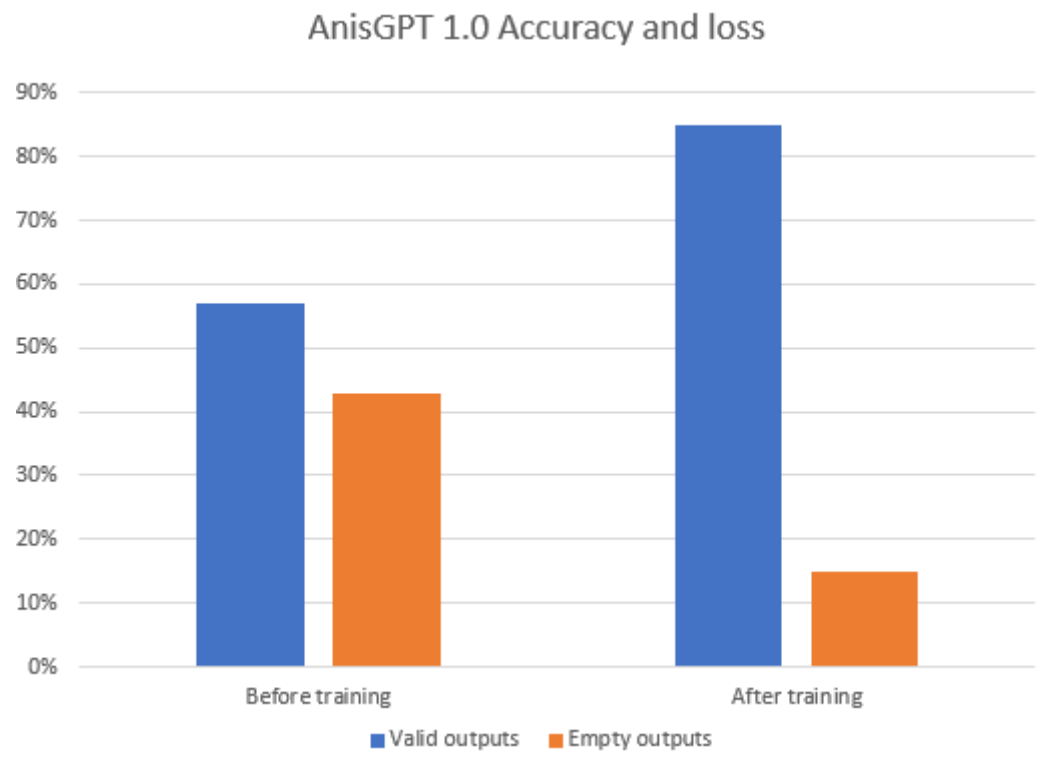

Figure III.29: ARW chatbot accuracy and loss rates.

III.3.2. ARW chatbot

III.3.2.1. Conception

A chatbot for booking train tickets can be a combination of both a rule-based and a machine learning model. Let me explain the distinction between the two:

- **Rule-based system**

A rule-based chatbot operates on predefined rules and logic. It follows a set of if-then statements or rules to generate responses. The initial part of our chatbot that asks the customer what they need and provides buttons for selection can be implemented using rules. For example, if the customer selects the "Book a Ticket" button, the chatbot can trigger a specific

set of rules to guide the conversation towards collecting the necessary information for ticket booking.

- **Machine learning model**

A machine learning-based chatbot leverages algorithms that learn from data to generate responses. In our case, once the chatbot collects the relevant information from the customer (such as station, time, and trip date), it can use a machine learning model to analyze and process the input. The model will utilize techniques like natural language processing (NLP) and dialogue management to understand the customer's intent and provide appropriate responses. This can involve techniques like named entity recognition to extract specific details from the customer's input, such as the station name or trip date.

To summarize, the initial part of ARW chatbot that presents options and collects customer preferences is rule-based, while the subsequent steps that involve understanding and processing the collected information can employ machine learning (ML) techniques. At the ending of this chapter, we will achieve a combination these approaches, you can create an effective chatbot that provides a seamless and personalized booking experience for your customers.

- ✓ **About ARW company**

ARW is an abbreviation of ALGERIAN RailWays, is a **FICTIOUS** company that I created in order the achieve my project goal.

ARW is a transportation company that operates high-speed TGV trains in Algeria. The company stations are situated in four (4) main Algerian cities: Boumerdes, Algiers, Annaba and Oran.

With over than 30 trips a day from and to all 4 destinations and the ability of transporting more than 1000 passenger a day means that it’s very important to improve our customer services, and so we developed an AI virtual assistant called: ARW Chatbot, using several tools from programming languages to chatbot’s building platform.

- ✓ At the beginning we created an account on (Chatbot.com) platform, after the log in successes, we go to the stories block (each project is called a story), and just by clicking on creating a new project, patterned blocks will show up just like the in figure below:

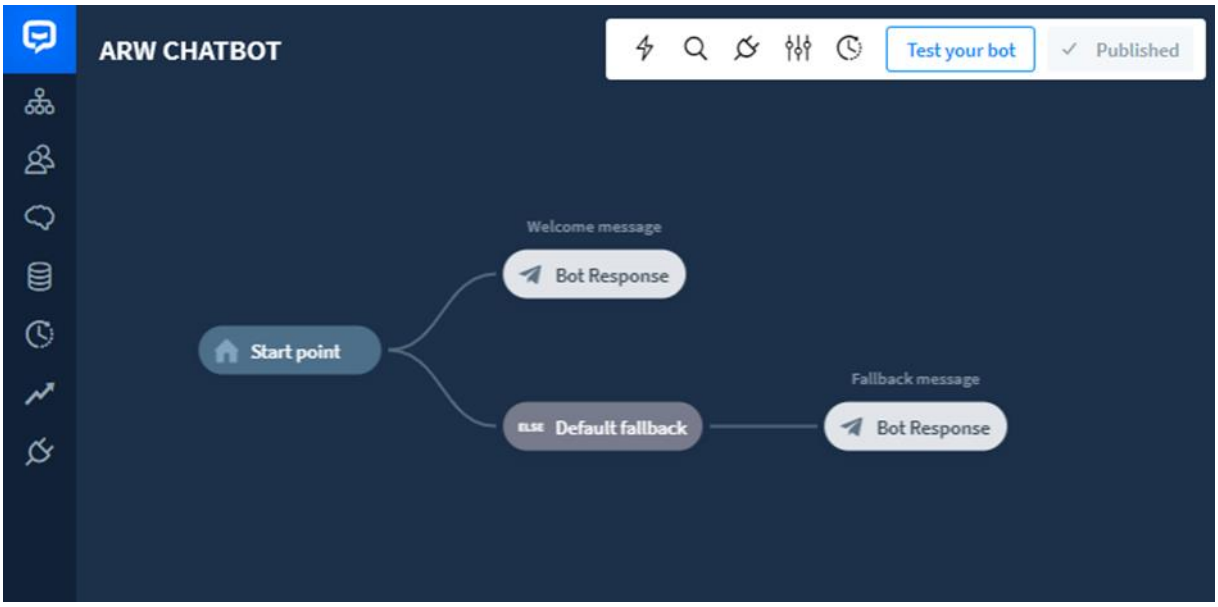

Figure III.30: First patterned story.

- ✓ Each block must be connected with another one or it will be ignored during the process.
- ✓ The main blocks are : Start point that is fixed, User input, Bot response, Default fallback, Questions, and the Chat's closing one.
- ✓ The first to block we started building is the 'Welcome Message' that comes just after opening the chat window.
- ✓ We add the welcoming messages we want salutate the customer with, and a list of 4 options, each one has its own pattern and task, as shown in the figure below:

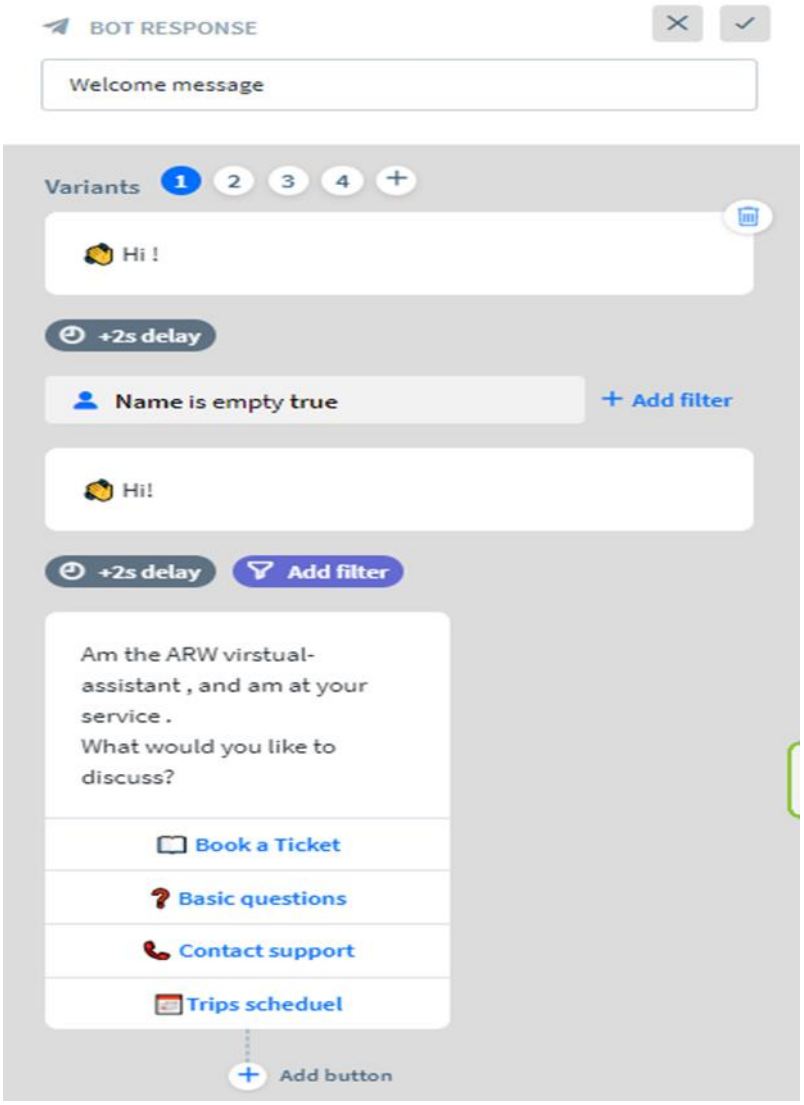

Figure III.31: Welcoming messages and tasks integration.

- ✓ From the end of the 'welcoming message' block, we start to extract and link 5 other blocks, but this time it's a user's input blocks, that means there are only 5 patterns the conversation takes after the welcoming message.

➢ **Booking a ticket**

- ✓ The first User's input block is named **Booking a ticket**. The conversation moves there when the user click on the Button associated.
- ✓ After that, our chatbot starts the ticket's reservation process, and begins to gather the costumer's informations including: Name, PhoneNumber and Email address.

The figure below shows how to set up these requests in the blocks:

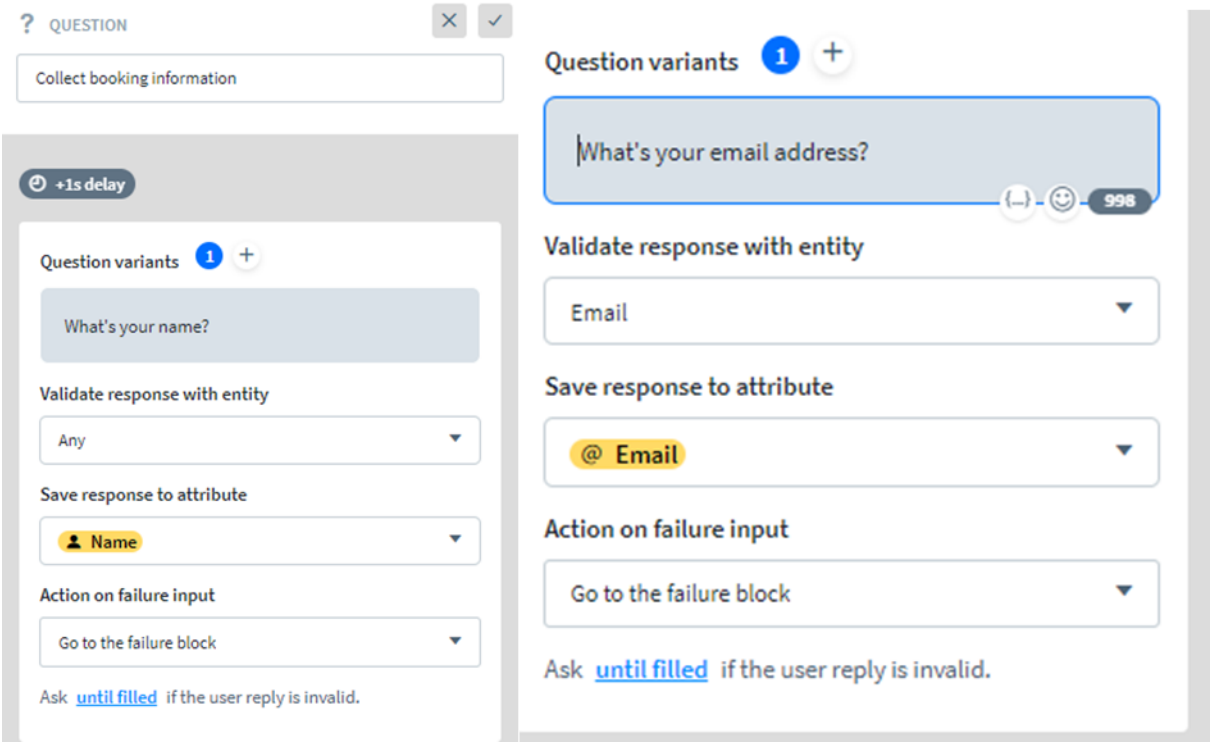

Figure III.32: Setting up the personal information gathering

After gathering the personal information's, each one of them gets registered as a specific attribute. For example, when the customer enters his email address it will be automatically registered in the {@Email} attribute. Same thing for the other informations.

- ✓ The second block that appears after writing the personal informations, is the Departure and Arrival stations, the bot asks the costumer from where he would like to book, and to what destination.
- ✓ This time, the user gets to choose One of four departure stations, so that the choice made is eliminated from Arrival destinations.

For example, if the user chooses Boumerdes as a departure station, he will have the right to choose only Annaba, Oran or Algiers, and Boumerdes will be eliminated from Arrival stations automatically.

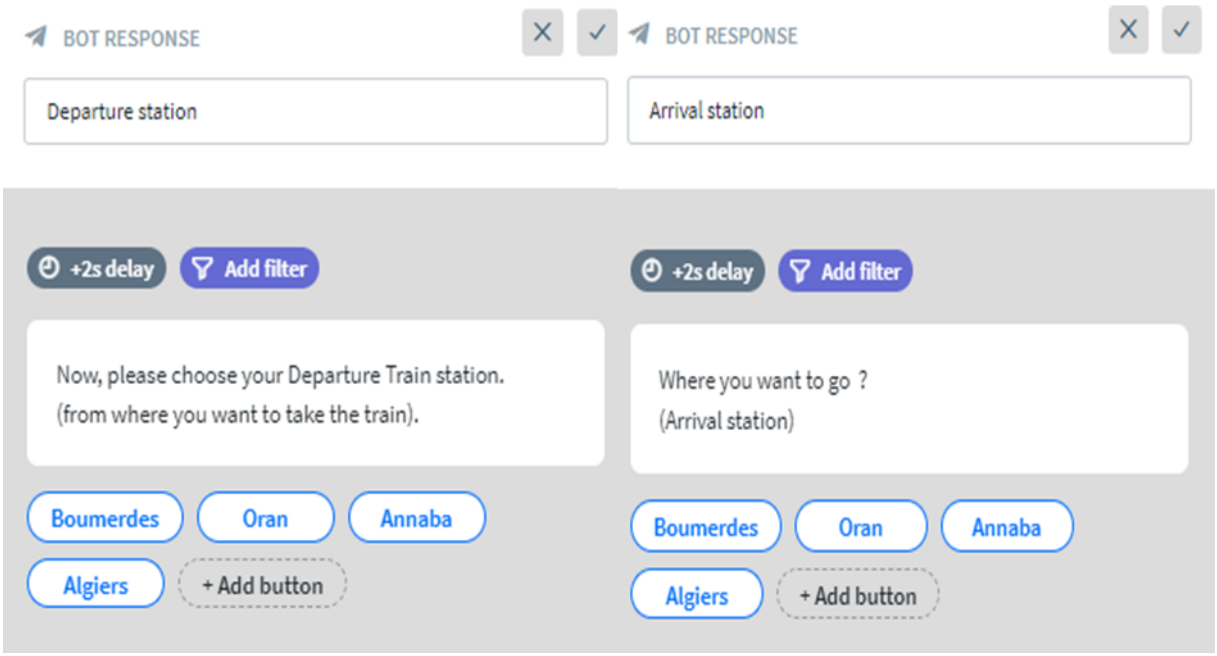

Figure III.33: Gathering the departure and arrival station

- ✓ After confirming the recent task, we will program the bot to ask the costumer how many tickets he wants to book.

What will happen at this point, is that the bot will book the number of tickets and every time the number is more than 1 [VALUE], the other tickets will be registered with the costumer's informations.

To explain more, if I booked 3 tickets using my name 'Anis', when we search the DATABASE, we will find 3 tickets booked by 'Anis' and with all the same attributes.

- ✓ The final step before confirming the booking is selecting the Departure times.

We set 6 different departure times, after the time's selection, its gets registered to the attribute named {departureTime}.

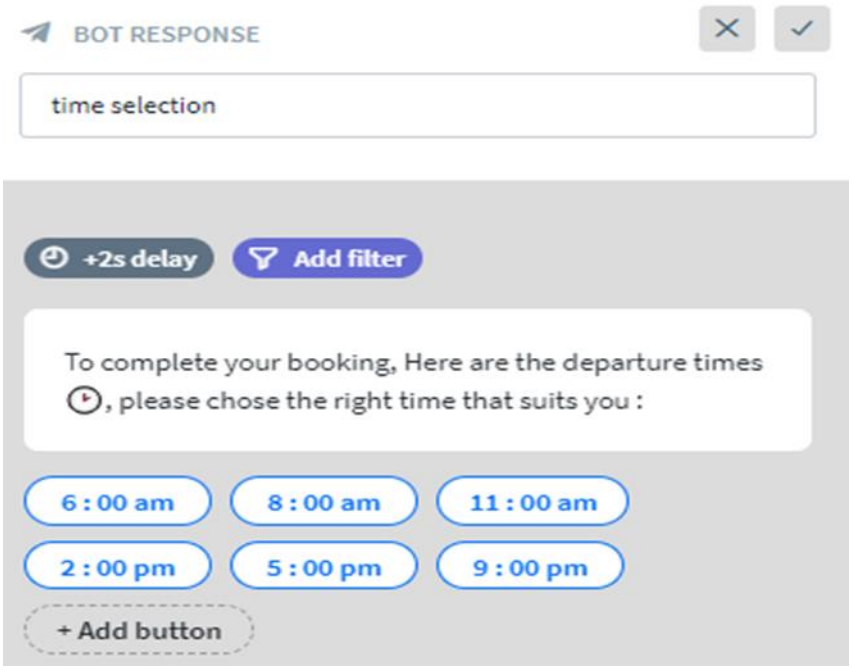

Figure III.34: Setting departure times

- ✓ Arriving to the confirmation step means that the bot has completed collecting the necessary booking information (Name, phoneNumber, Email, Number of tickets, Departure station, Arrival station and Departure time) and it will be stored as attributes to be used later in the DATABASE collection.
- ✓ A confirmation message will pop up including these collected informations and two buttons as shown in figure III.35.

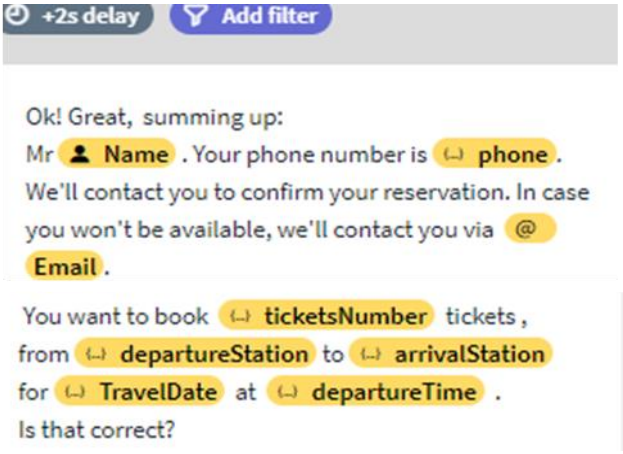

Figure III.36: Setting the confirmation message.

✓ After the confirmation message in sent, the costumer will have two choices, YES or NO buttons, if he chooses YES the confirmation is done and the reservation was made by success.

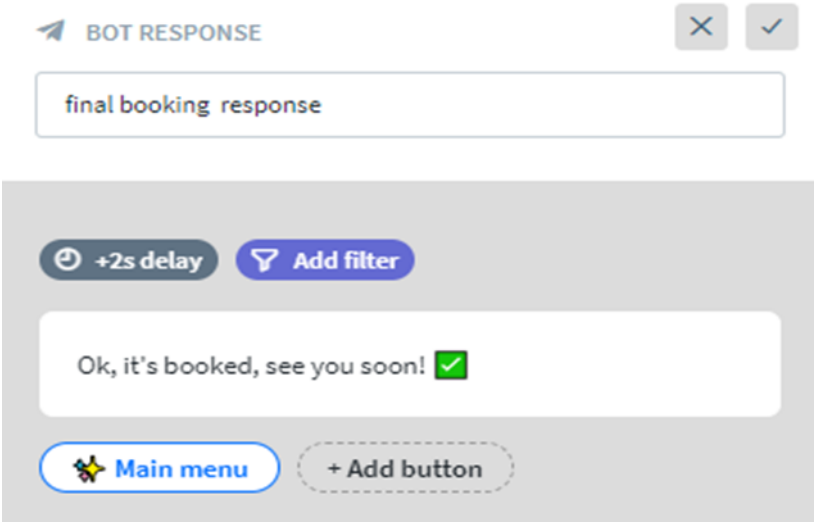

Figure III.37: Setting up the Success booking message

✓ But if the costumer chooses No, the bot redirects him to the main menu and ask for what it can help with.

➢ **Basic questions**

✓ The second pattern set is called Basic question, and it's block is called when the customer asks for help or for general informations.

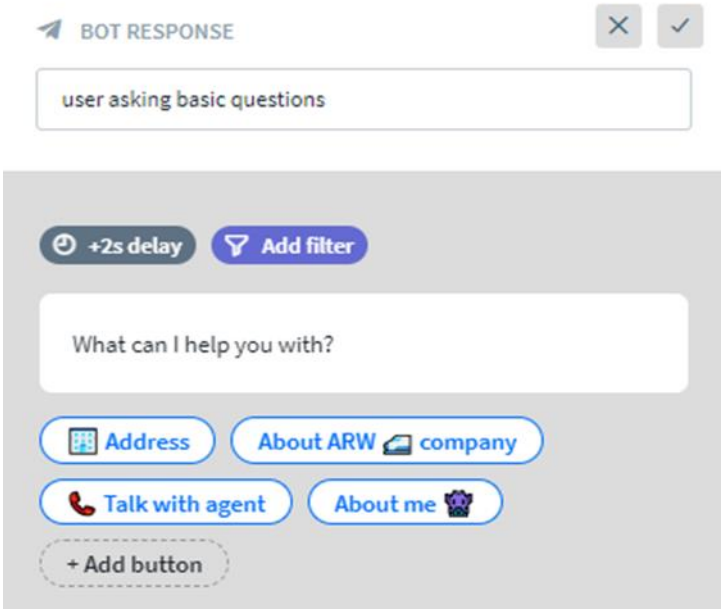

Figure III.38: Bot offering help message.

✓ This section contains 4 main subsections, the first one is about ARW company, here the bot replies with a standard message:

(ARW, an abbreviation of "ALGERIAN Railways", is a public-transport company, with over than 30 trips a day, and transporting more than 1000 passenger per day. For now, we are available in only 4 main Algerian states (BOUMERDES, ALGIERS, ORAN, ANNABA), and we are on the right way to add other destination at the up-coming period).

✓ The second section is about the bot itself, so it replies with this message:

(i am a Chatbot named "ARW virtual assistant", i was developed by the ARW software engineer "ANIS BOUGUELIA" at the ending of 2022, and i was deployed on service since the 1st of March 2023. I was made to help our customers with finding and booking train tickets with ARW company the best way it could be).

✓ The third one is about the company or station's locations, I set up the Address block with integrating a google Map link that shows up when asking.

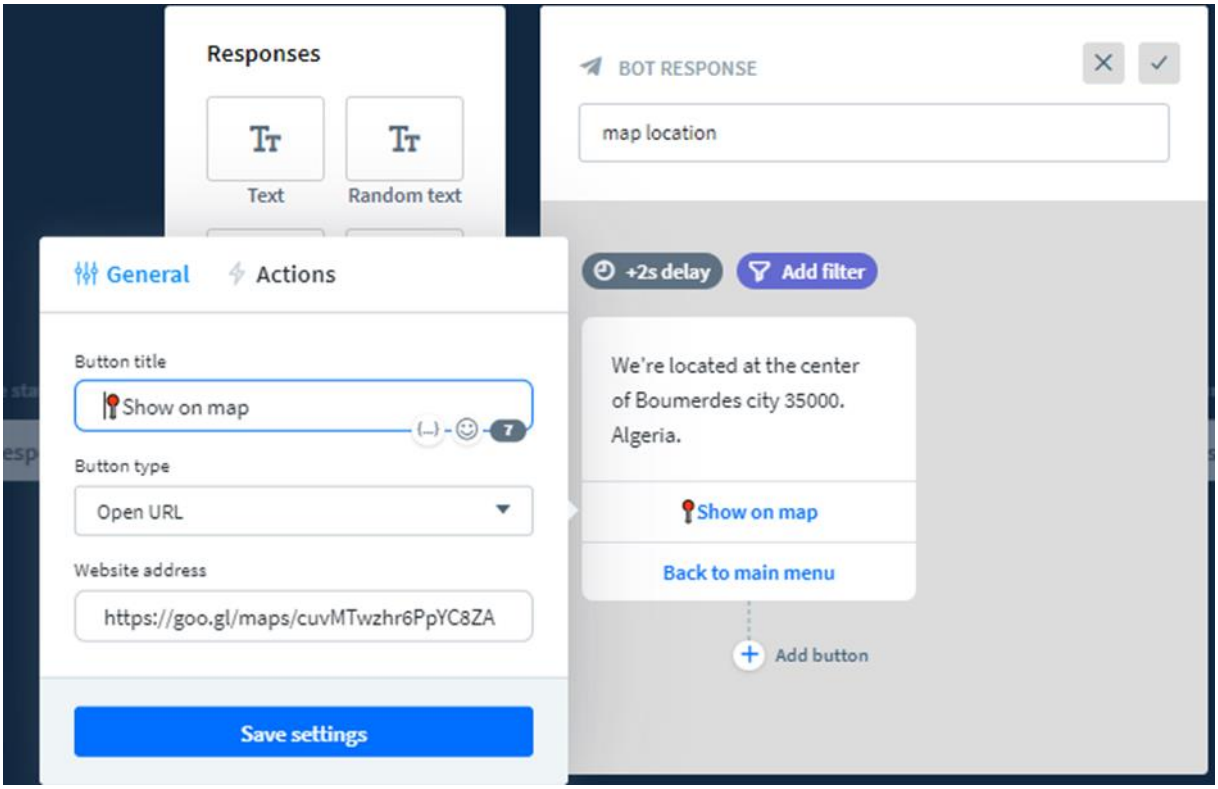

Figure III.39: Setting up the ARW Map location.

These two subsections mentioned bellow, can be called with several inputs, so we added them and trained our bot to get usual to these inputs and recognizes them next time.

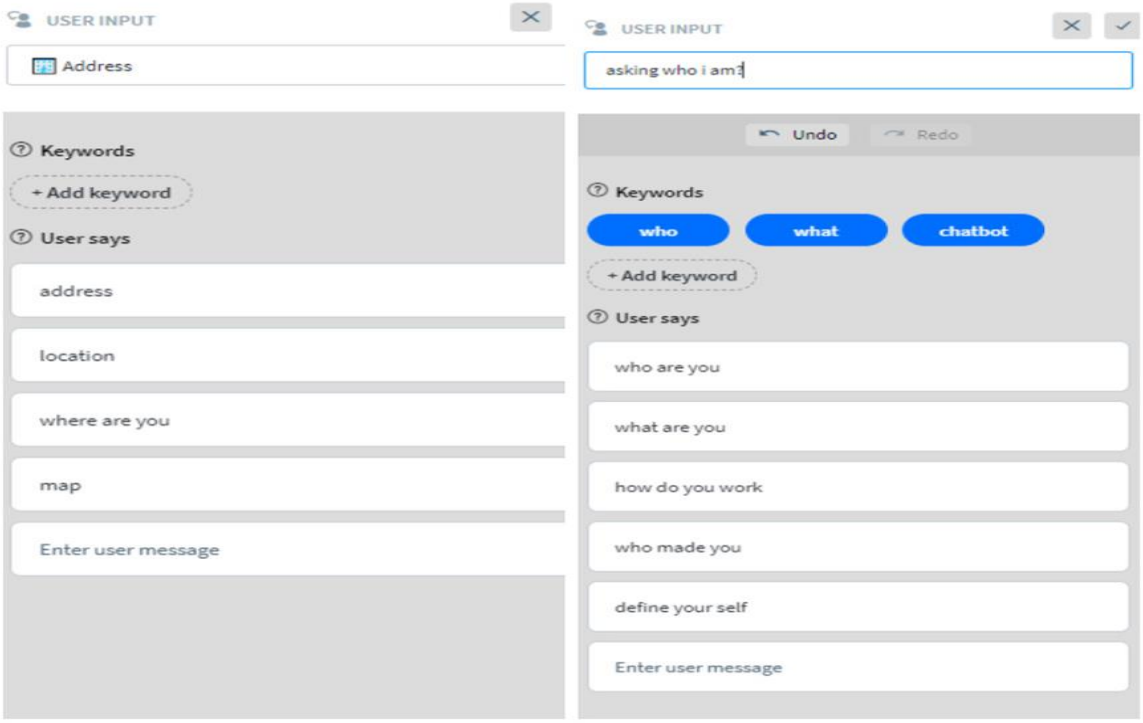

Figure III.40: Adresse and Identity inities.

- ✓ These keywords and sentences are stored after the users input into the database, so that our Machine learning model will be capable of dealing with tasks later in less than the past time.

- **Contact support**

To call this block, the user has to input a list of word or sentences called entities, the bot is trained before on these entities and it will understand directly that the customer is asking for Support contact.

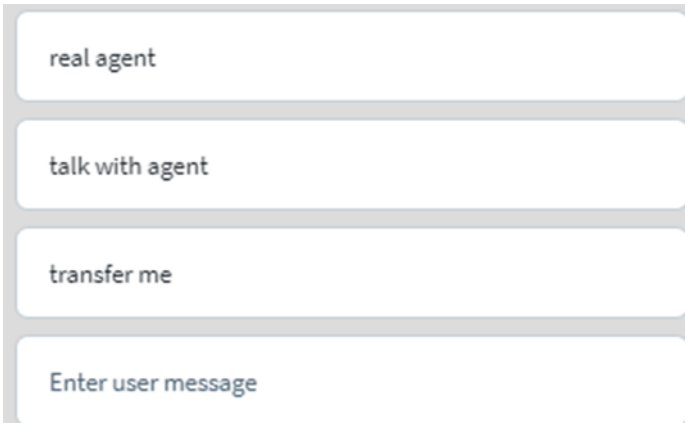

Figure III.41: Asking for support entities

After the user requests contact support, the bot automatically answers him by this standard message:

(Please contact us at 0558195598 or send us an email to support@arw.dz.

we'll be happy to help you!)

- **End chat**

This cell is considered necessary in out chatbots architecture, it's responsible of closing and ending the chat. The entities of calling this cell are the common closing words end goodbyes like: Bye, close chat, take care, goodbye…

The bot responds with saying: See you later and the chat gets ended and closed.

- **Trips schedule**

In this block, we stored the Schedule of our train trips, including Departure station, arrival station and Departure time. The figure bellow shows how to integer any trip information into this cell:

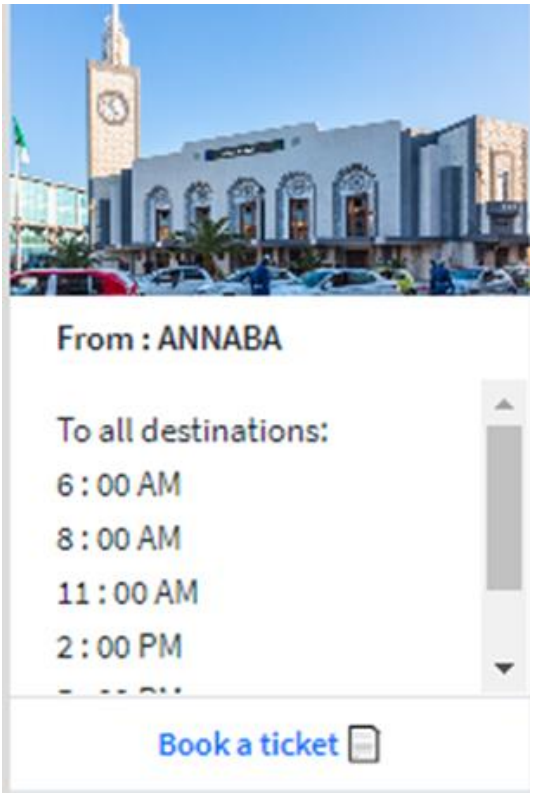

Figure III.42: Setting up the ARW Map location

III.3.2.2. Results

After around 3 weeks of training our Chatbot on handling all the tasks that I programmed it for, the bot reached around 90% of Valid tasks and the 10% left were some breakdowns such as unrecognized entities, and some server bugs.

However, after these 3 weeks I reached around all four main tasks and the bot interacted with me correctly.

- **The user's interface**

The testing phase was made on a predefined chat window over a hosting website offered by the (Chatbot.com) platform. The figure bellow shows how our website looks like:

Figure III.43: ARW website with Chatbot window.

This window contains 4 main components:

- The “start the conversation” button.
- The chatbot’s name.
- A quick short welcoming message.
- 3 external links, that contains the Social media contacts, profiles, and more. of the ARW company.

Once the user click on ‘Start the conversation’, the welcoming message shows up, which means that our Chatbot interacted with the user’s first request.

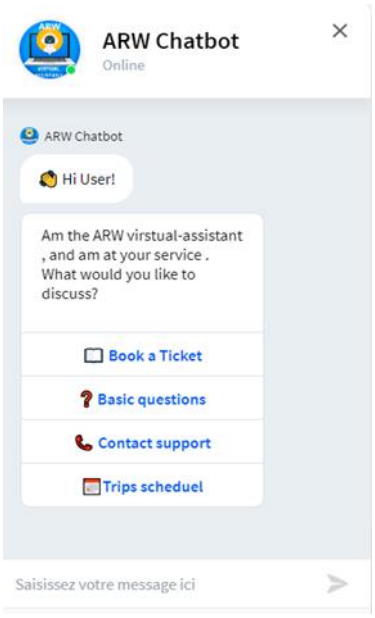

Figure III.44: First welcoming message after starting the chat.

- Our project objective is to build a Booking train ticket Chatbot, it means to book we got to click on the first Button: Book a ticket, after that the bot starts asking the user on his personal informations, as shown in the figure below:

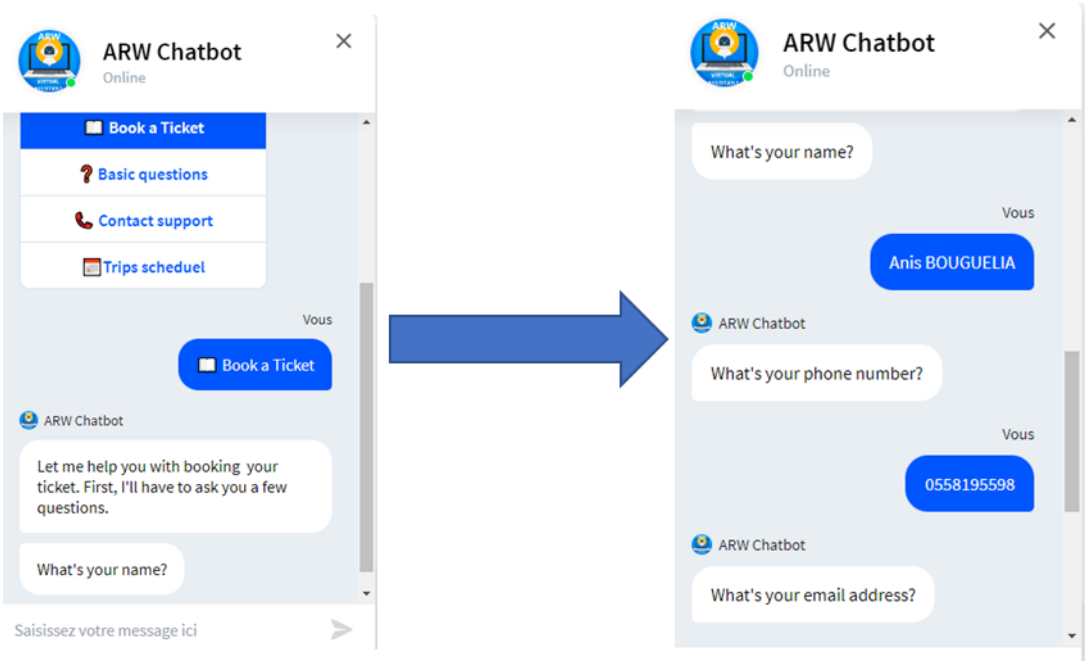

Figure III.45: Collecting personal informations.

- After that, The bot starts asking about The booking details such as: Destination, Departure station, departure's date and time plus the tickets number.

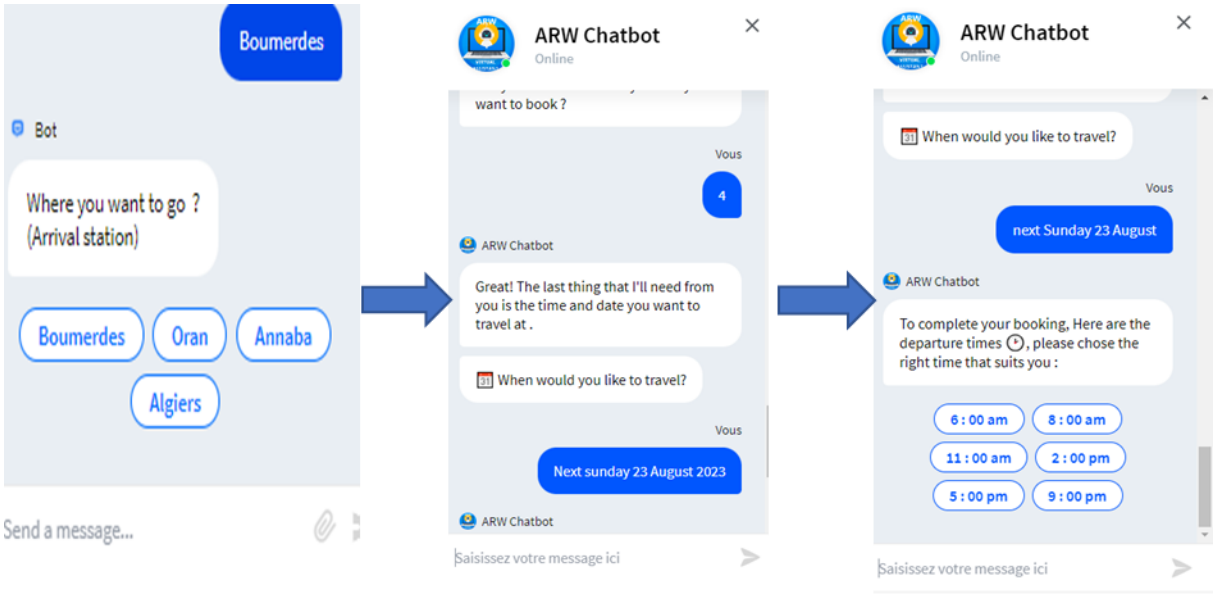

Figure III.46: Bot asking about Booking details.

- Arriving here means that the reservation process is almost done, all what is left is the confirmation step, where the bot gathers all your information and replies you with a confirmation message request that contains all your personal information's and the Booking schedule.

Confirmation message that contains all the booking informations.

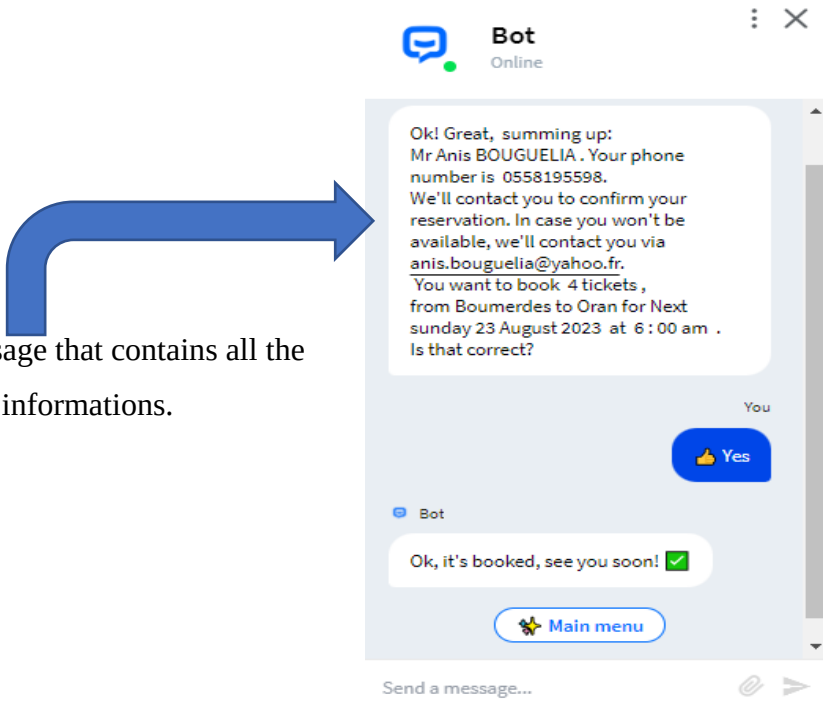

Figure III.47: The booking's confirmation Message

- The reception of the booking's confirmation message means that your booking has been confirmed and your tickets are now yours, all you've left is making the purchase.

- **Attributes**

Now, we will talk about the underlaying process results, those that contains the values stored after collecting it from then user, named **{} attributes**.

These attributes are stored on the platform's cloud, and it can be used as a data in our database to train our model next time, that means the more customers book tickets the more our chatbot become stronger and more structured.

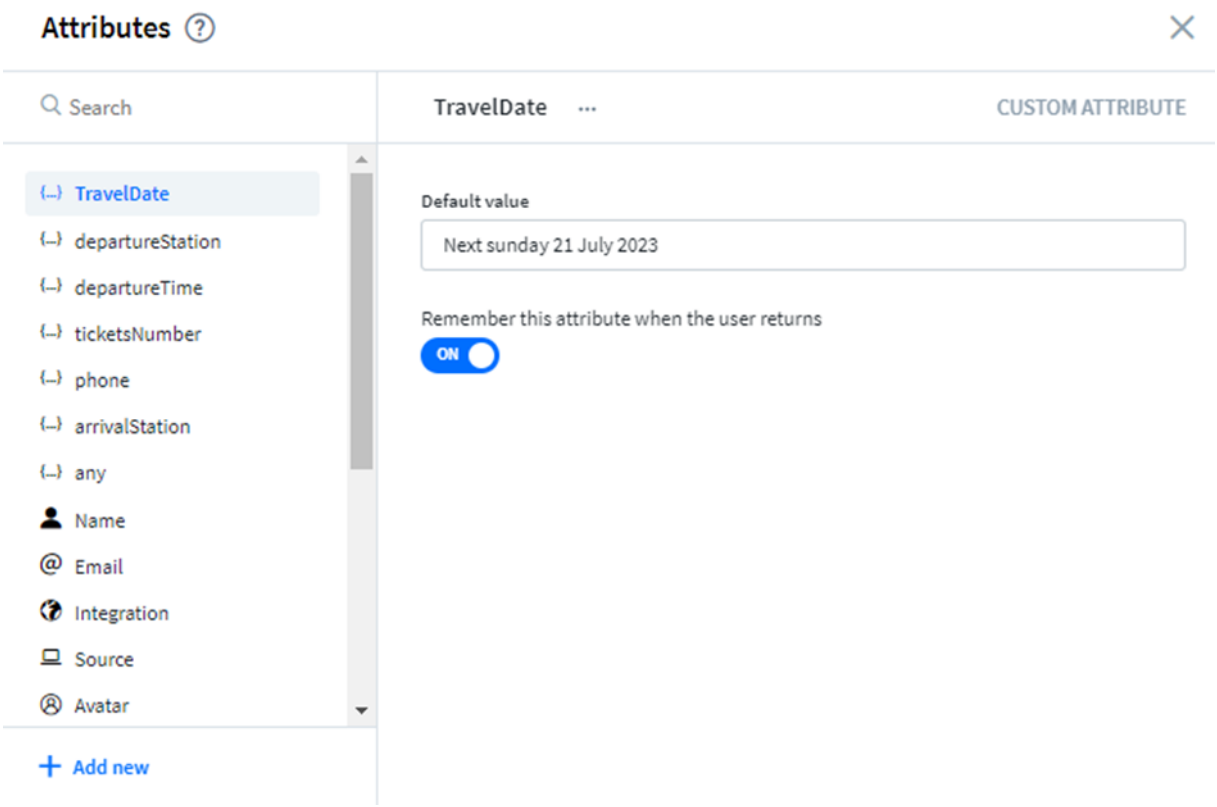

Figure III.48: Attributes list.

- You can see here that the attributes collected from the user's interactions are stored automatically in the database server, so that our model use it as training data.
- One more thing, is that even if the users interprets the conversation flow of he closes the chat window directly, the attributes collected by our system get stored too into the database, no matter how correct or wrong it is.

- **JSON responses**

JSON (JavaScript Object Notation) is a lightweight data format commonly used to store and exchange structured information. It is widely utilized in web development and is particularly useful for transmitting data between a client (such as a my chatbot) and a server.

In the context of my train ticket booking chatbot, JSON can be employed to capture and organize the user's information during the booking process. When the user provides their name, email, phone number, destination, arrival station, and hour, this data can be stored in a JSON object.

Here's an example of how the user's booking informations are represented in JSON format:

```json
{
  "name": "Anis BOUGUELIA",
  "email": "anis.umbb@gmail.com",
  "phone": "0558195598",
  "destination": "Boumerdes",
  "arrival_station": "Oran",
  "hour": "2:00 PM",
  "number_of_tickets": 4
}
```

Figure III.49: Booking informations in JSON format.

- **The booking informations**

After the ticket's booking process is done, and confirming the booking by the user, his data goes directly to the server in JSON format to be registered and used after as a training data in the chatbot's database.

The figure bellow shows how the booking's data are stored and saved in the server:

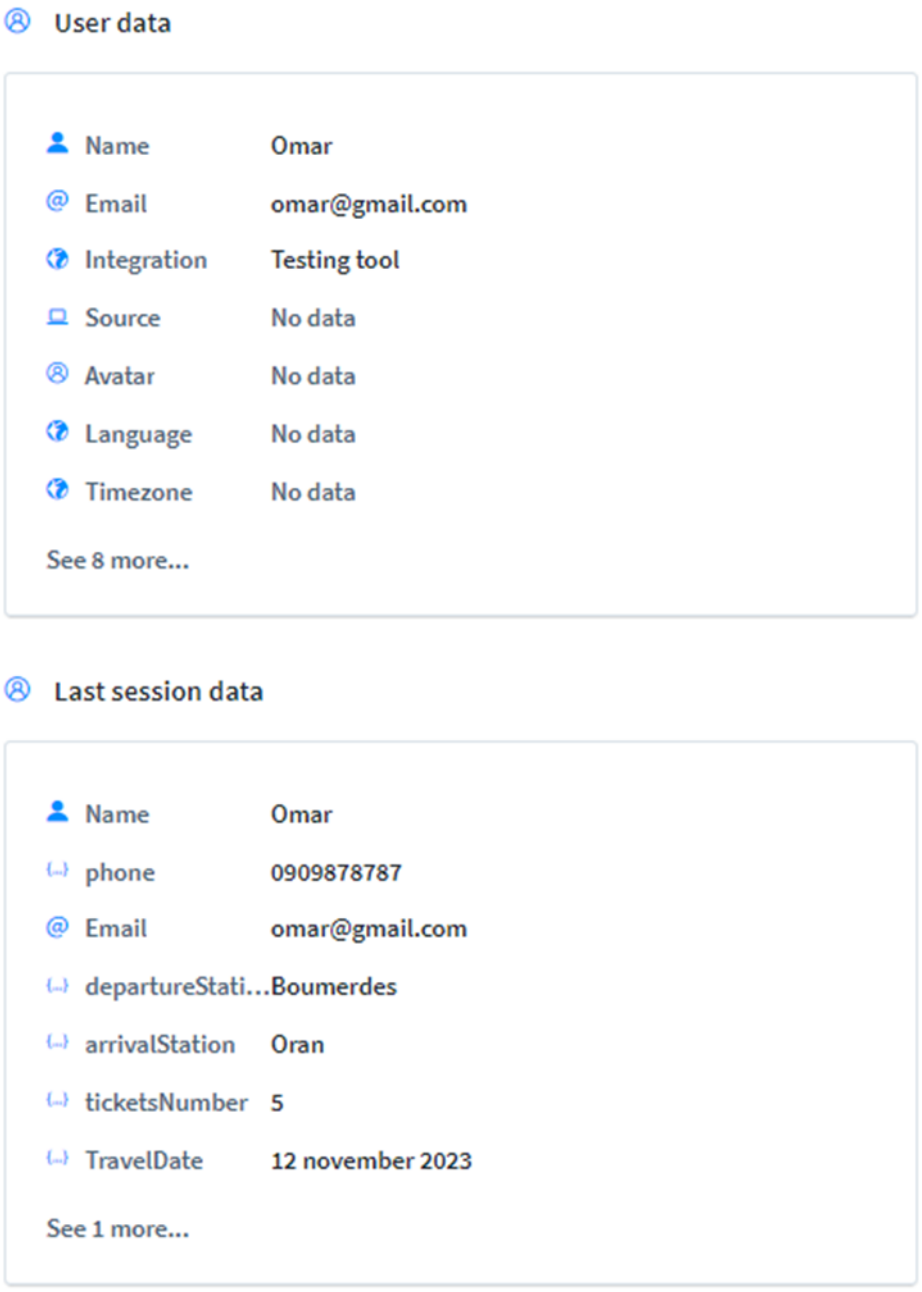

Figure III.50: User's data after a successful booking.

III.3.2.3. Results discussion

At the beginning of the training session, the bots showed a low rate of accuracy, I could achieve only 33 successful bookings, and that meant that I should train my chatbot more to improve this accuracy rate.

The ARW chatbot has shown perfect results after the training session, with training it over than 100 time (I made 100 booking requests), 95 of them were successful requests and only the 5 others left failed during the process because of the lack of training data.

But the fact that our model achieved 95% of validity, means that it's a well-structured chatbot, and ready to be deployed.

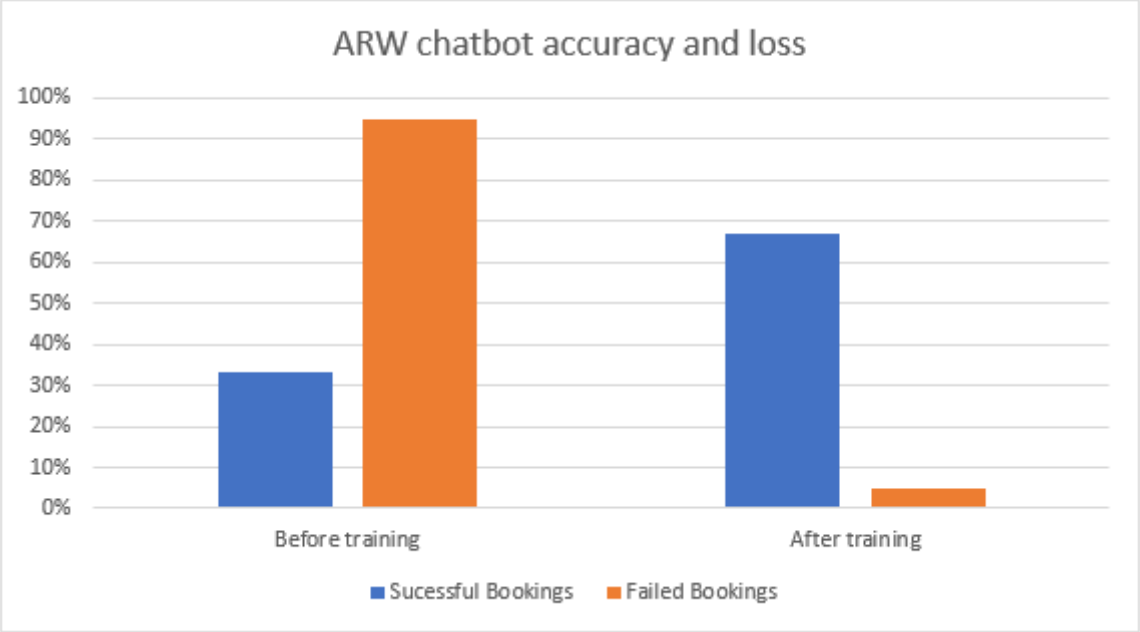

Figure III.51: ARW chatbot accuracy and loss rates.

III.4. Conclusion

In conclusion, the building and implementation of the chatbots was a success. The initial chatbots (AnisGPT 1.0) based on API's showed an accuracy rate of 85%, the second chatbot (ARW chatbot) that was a built by 2 approaches (rule-based and machine learning) resulted in a 95% accuracy rate.

These results indicate that the chatbot is ready to be used and can effectively handle user requests with a high degree of accuracy. Overall, the building and implementation of the chatbot has proven to be a valuable tool for providing efficient and effective customer service.

General conclusion

General conclusion

In conclusion, this thesis emphasizes the crucial role of artificial intelligence in addressing the evolving needs of various domains. With a specific focus on building a customer services chatbot for booking train tickets, the combination of rule-based and machine learning approaches offers a superior solution for train companies, significantly enhancing the quality of customer services.

The integration of artificial intelligence in the form of this chatbot serves as a positive changes and improvements. By automating and personalizing the ticket's booking process, the chatbot provides a seamless and efficient experience for customers. This streamlined approach not only saves time but also enhances overall customer satisfaction and loyalty.

Furthermore, the implementation of the chatbot offers concrete benefits for train companies. Through its intelligent algorithms and automated assistance, the chatbot reduces the workload on customer service representatives, allowing them to focus on more complex inquiries. This improved efficiency not only boosts the businesses productivity but also contributes to cost savings.

In summary, the development of this customer services chatbot for booking train tickets highlights the indispensability of artificial intelligence in meeting the demands of today's dynamic market. By leveraging a combination of rule-based and machine learning approaches, the chatbot presents an optimized solution for train companies, revolutionizing customer services, and boosting business growth. Its deployment promises to elevate customer satisfaction, improve operational efficiency, and position train companies at the forefront of innovation in the industry.

Future vision

Future vision

After successfully deploying my train ticket reservation chatbot and witnessing its exceptional results, my future work will revolve around further enhancing customer services within the trains and railways industry, ensuring the utmost satisfaction of clients. Building upon the success of the chatbot, I intend to delve deeper into understanding the specific pain points and challenges faced by customers when it comes to train travel.

By conducting thorough research and analysis, I aim to develop innovative solutions that streamline the customer experience, optimize ticketing processes, and provide personalized assistance throughout the journey. Through the integration of advanced technologies such as natural language processing and machine learning, I envision creating a comprehensive customer service ecosystem that not only facilitates ticket reservations but also offers real-time information updates, efficient grievance redressal mechanisms, and tailored recommendations based on individual preferences. Ultimately, my goal is to revolutionize the way customers interact with trains and railways, making their journeys seamless, convenient, and ultimately, delightful.

Bibliography

[1] Russell, S., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach (3rd ed.). Pearson.

[2] Nilsson, N. J. (2014). Artificial Intelligence: A Guide to Intelligent Systems. Morgan Kaufmann.

[3] Mitchell, T. (1997). Machine Learning. McGraw-Hill.

[4] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[7] Loenen, E. Peters, R. (2019). The Evolution of Chatbots: A Literature Review of Technologies, Techniques, and Application Areas. In 2019 8th International Conference on Software and Computer Applications (ICSCA) (pp. 334-339). IEEE.

[8] Levin, M. E. (2018). Conversational Agents in Electronic Commerce: A Brief History and Framework. In Advances in Electronic Commerce, Web Application and Communication (pp. 15-29). Springer.

[9] Picazo-Vela, S., Gutiérrez-Martínez, I., Luna-Reyes, L. F. Zavala-Cruz, J. (2012). Understanding risks, benefits, and strategic alternatives of chatbots implementation in organizations. Computers in Human Behavior, 28(5), 1666-1673.

[10] McTear, M., Callejas, Z. Griol, D. (2016). The Conversational Interface: Talking to Smart Devices. Springer.

[11] Hutto, C. J. Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth International Conference on Weblogs and social media (pp. 216-225).

[12] Zhang, Y. Zhang, D. (2019). Chatbots in education: A review of recent advances and future prospects. Journal of Computing in Higher Education, 31(2), 262-279.

[13] Pratama, A. Y. Rachmawati, L. (2019). A review of chatbot technology. Journal of Physics: Conference Series, 1365(1), 012050.

[14] Ahamed, S. I., & Hasan, M. S. (2020). Chatbot technology: Classification, application, tools, and challenges. In Innovations in Bio-Inspired Computing and Applications (pp. 381- 390). Springer.

[15] Dahiya, S. Yadav, S. K. (2020). A comprehensive study on chatbot architecture. In Proceedings of 5th International Conference on Internet of Things: Smart Innovation and Usages (pp. 415-420). Springer.

[16] Lathia, S. Bhatia, A. (2021). Designing a scalable, modular and efficient chatbot architecture for industry automation. In 2021 4th International Conference on Advanced Technologies in Signal, Image and Video Processing (ATSIV) (pp. 1-5). IEEE.

[17] Akerkar, R. (2019). A beginner's guide to chatbots. In Practical Artificial Intelligence for Cybersecurity (pp. 1-13). Apress.

[18] Gao, L., & Sun, W. (2020). A literature review of chatbots in education: Challenges and possibilities. Journal of Educational Technology Development and Exchange, 13(1), 101-121.

[19] Luger, E., & Sellen, A. (2016). Like having a really bad PA: The gulf between user expectation and experience of conversational agents. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 5286-5297). ACM.

[20] Karimi, A., & Khanlari, A. (2020). A Comparative Study of Rule-Based and Machine Learning-Based Sentiment Analysis Approaches. International Journal of Information Technology & Decision Making, 19(3), 915-944.

[21] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR).

[22] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. International Conference on Machine Learning (ICML).

[23] Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable Artificial Intelligence: Understanding, Visualizing, and Interpreting Deep Learning Models. arXiv preprint arXiv:1708.08296.

[24] Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for Interpreting and Understanding Deep Neural Networks. Digital Signal Processing, 73, 1-15

[25] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent Advances in Convolutional Neural Networks. Pattern Recognition, 77, 354-377.

[26] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2019). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. Advances in Neural Information Processing Systems (NeurIPS).

[28] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[29] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (pp. 4171-4186).

[30] Ruder, S., & Howard, J. (2018). Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 328-339).

[31] Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1870-1879).

[32] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1480-1489).

[33] See, A., & Liu, P. J. (2017). Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1073-1083).

[34] Guo, J., Xu, W., Yang, G., & Li, H. (2020). Neural Text Generation: A Practical Guide. Synthesis Lectures on Human Language Technologies, 13(3), 1-211.

[35] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2019). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[36] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Webography

[W1] **https://www.ibm.com/topics/deep-learning** [Accessed on April 25, 2023]

[W2] **https://www.turing.com/kb/how-neural-network-models-in-machine-learning-work** [Accessed on April 25, 2023]

[W3]**https://towardsdatascience.com/a-beginner-friendly-explanation-of-how-neural-networks-work-55064db60df4** [Accessed on April 25, 2023]

[W4]**https://levity.ai/blog/neural-networks-cnn-ann-rnn#:~:text=ANNs%20(Artificial%20Neural%20Networks)%20are,proficient%20in%20Natural%20Language%20Processing.** [Accessed on April 25, 2023]

[W5] **https://levity.ai/blog/how-natural-language-processing-works** [Accessed on May 05, 2023]

[W6]**https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python** [Accessed on May 17, 2023]

[W7] **https://www.python.org/** [Accessed on May 17, 2023]

[W8] **https://colab.research.google.com/** [Accessed on May 19, 2023]

[W9] **https:// app.chatbot.com/dashboard/64a9bfd8657b350007bfd5a9/** [Accessed on June 05 , 2023]

[W10**]** **https://code.visualstudio.com/** [Accessed on June 02, 2023]

[W11] **https://realpython.com/nltk-nlp-python/** [Accessed on May 17, 2023]

[W12]**https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON** [Accessed on June 19, 2023]