الجمهورية الجزائرية الديمقراطية الشعبية
The People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة محمد بوضياف بالمسيلة
University Mohamed Boudiaf of M'sila

كلية الرياضيات والإعلام الآلي
Faculty of Mathematics and Informatics

قسم الإعلام الآلي
Department of Computer Science

**Domain:** Mathematics and Computer Science

Thesis Presented to Fulfill the Partial Requirement
For Master's Degree in Computer Science

**Specialty:** Information System and Software engineering.

**Prepared By:** Achwak Berrah.

**Supervised By:**

Said Gadri.

ENTITLED

# An Academic Chatbot Using NLP and Deep Learning.

**Jury Members**

| | |
|---|---|
| Hamza Loucif | President |
| Said Gadri | Supervisor |
| Mohamed Bahache | Examiner |

**Academic Year 2023/2024**

# الإهداء

- إلى سَيَّدي و حِبّي رسول الله المجاهد الشهيد المثخن في أعداء الله صلوات ربي وسلامه عليه.
- إلى الذين يمشون على أثره ويصنعون لنا معاني عِزِّ الاسلام، رجالُ العبور المقدس الأطهار الأبرار، أحفاد الياسين وصُنَّاعه من رجل الظل أبو خالد الضيف إلى الإعلام العسكري والمكتب السياسي وكل من شدّ على أياديهم .
- إلى المُلثّم بكوفية العزّ الحمراء –لا حرمنا الطلّة– سمعتك حين ناديت "فليجدكم العدو حيث يحذر "فأجبت النداء وقلت أنا لكم الفداء وتحرّيت الاتقان لله ولكم.
- إلى شيخي صالح الذي قال "سنمحو فكرة الكيان الصهيوني من عقولكم جميعا" أنت صدّقتها بالدم – تقبلك الله– وانا صدّقتها بهذا العمل وأقول "نمضي بما علمتنا يا صاحب الأمر ".
- إلى العالم الأسير الحرّ مُرعب الكيان أمير الظلّ عبد الله البرغوثي وجميع أسرى السجون والأنظمة فكّ الله بالعزّ أسرهم .
- إلى شهداء حرب السابع من اكتوبر المجيد ضحايا العدو الصهيوني النازي أطفالا ورجالا ولا أقول نساءً بل أقول صانعات المجد، شهداء مجزرة مستشفى المعمداني والشفاء ومجازر المساعدات ومجزرة جباليا ومجازر رفح وكلّ شهداء الأرض، ملتقانا في الجنّة عند حوض حبيبنا محمد صلى الله عليه وسلم، وقد ربح بيعكم أيها الأطهار.

روحي لكم الفداء يا من أتعبتم من بعدكم.

- إلى العائلة التي علّمتني حبّ فلسطين أبي وأمّي والإخوة الكرام وأصغر مُقاوِمة سُكّرة البيت جنان.
- إلى كل مسلم يحاول، ينهض ثم يتعثّر ثم ينهض من جديد هذا جهد المُقِلّ ،نصنعه لك ، ونضعه بين يديك...

والعلم إن لم يعد بالخير على الدين والمسجد الأقصى فلا خير فيه

# تشكرات

**الحمد لله الذي كتب لي تمام هذا العمل لقد استشعرت لطفه الخفي في كل التفاصيل ولأن من لم يشكر الناس لم يشكر الله أتقدم بالشكر الكثير للأستاذ المربي سعيد قادري الذي تعلّمت منه القيم قبل العلم له جزيل الشكر لأنّه زرع في داخلي معنى أن يكون الإنسان رساليا بعلمه متقنا وجاداً وأميناً ما زرعه في من أفكار تعجز عن زرعه المحاضرات والكتب أدام الله له عافتيه وحفظه بحفظه ونفع به وأدامه مثالا لنموذج الدكتور الرسالي القدوة كما أشكر كل من الأستاذ حمزة لوصيف والأستاذ بحاش محمد على جهدهم في تصحيح هذه المذكرة .**

**أشكر الأستاذة الصديقة سلمى على نصحها ودعمها وتشجيعها فلطالما عملت على رفع همتي وعزيمتي كلما فترت .**

**لكلّ من كان له دعم في هذا العمل لكم مني خالص الشكر والدعاء لا حرمت قربكم ودعمكم ونصحكم.**

# Table of Content

Table of Content....i
List of Figures …....iv
General Introduction....1

**CHAPITRE 01 :Enhancing University Efficiency with Chatbots.**

1. Introduction....5
2. Boosting University Efficiency with Chatbots....5
3. Faculty of Mathematics and Informatics....6
   3.1 Structure (Departments.)....6
   3.2 Challenges of The MI Faculty....6
   3.3 Main Faculty Administration Tasks....7
4. How Can We use AI to Facilitate Administration Tasks?....8
   4.1. Advantages of a Chatbot as a Faculty Administrator....9
   4.2. Disadvantages of a Chatbot as a Faculty Administrator....9
5.Conclusion:....10

**CHAPTER 2 :Chatbots Foundations and Approaches**

1. Introduction:....12
2. What is a Chatbot?....12
   2.1. Chatbot technology:....12
   2.2. Types of Chatbots:....13
   2.3. General Chatbot architecture:....15
3. Natural Language Processing (NLP)....15
   3.1. Definition:....15
   3.2. Natural language Processing Techniques and Methods....16
   3.3. NLP Applications....18
4 . Combining Deep Learning and NLP:....19
   4.1 Convolutional Neural Networks (CNNs):....19
   4.2. Recurrent Neural Networks (RNNs):....20
   4.3. Long Short-Term Memory LSTM....21
   4.4. Auto Encoder:....21
   4.5. Transformers....22
   4.6. Generative Adversarial Networks (GANs)....23
5. Other Models and Techniques to perform NLP Tasks....24

5.1. Transfer Learning and Pre-trained Language Models: ....24
5.2. Word Embedding: ....25
6. Examples of Existing Chatbots....26
6.1. SIRI....26
6.2. ALEXA ....27
6.3. ChatGPT ....28
6.4. RASA....28
7.Conclusion ....29

**CHAPTER 3: IMPLEMENTATION AND REALIZATION**

1. Introduction: ....12
2. Used tools: ....12
2.1.Anaconda: ....12
2.2 Python: ....12
2.3TensorFlow: ....12
2.4. Keras: ....13
2.5. JSON (JavaScript Object Notation): ....13
2.6. Spyder: ....13
2.7. UTF8Encoding: ....13
2.8. Jupyter Notebook: ....14
2.9. Notepad++: ....14
2.10. Numpy: ....14
2.11. Vscode (Visual Studio Code): ....14
2.12. NLTK (Natural Language Toolkit): ....15
2.13. Pandas: ....15
2.14. Seaborn: ....15
2.15. Matplotlib: ....15
2.16. Scikit-Learn: ....15
3. Methodology Explanation: ....15
3.1. Performing the knowledge data set....16
3.2. Saving the Knowledge Base ....17
3.3. Finding the Best Match....17
3.4. Retrieving the Answer ....18
3.5. Main function to handle user input and respond ....18
3.6. Learning the Chatbot to Answer a New Question ....19

3.7. Main Execution....20
4.Chatbot Interaction....20
5. Explanation of Interfaces :....22
5.1. Interactivity:....22
5.2. Using other Languages....23
5.3. Comprehensiveness:....24
6. Suggestions for Improvement....25
6.1. User interface:....25
6.2. Expanding the knowledge base:....25
6.3. Improve the interaction:....25
6.4. Create an interactive version on smartphone:....25
6.5. Voice Recognition and Voice Answering Technology:....25
7. Kinds of Question/Answers Processed by UnivBot....26
8. Conclusion:....27
General Conclusion....28
Bibliography ….....30
Abstract …….....32

# List of Figures

Figure 1. Classification of Chatbots. ....13
Figure 2. General Chatbot architecture. ....15
Figure 3. Convolutional Neural Networks (CNN) and layer types. ....20
Figure 4.The Recurrent Neural Networks. ....20
Figure 5.Long Short-Term Memory LSTM ....21
Figure 6. The Auto Encoder. ....22
Figure 7. Transformers ....23
Figure 8. Generative Adversarial Networks (GANs) ....24
Figure 9. Transfer Learning and Pre-trained Language Models ....25
Figure 10.GloVe ....26
Figure 11. Word2Vec. ....26
Figure 12.Siri. ....27
Figure 13. ALEXA. ....27
Figure 14. ChatGPT ....28
Figure 15. RASA ....29
Figure 16. Knowledge Base. ....35
Figure 17. Save the updated knowledge base to the JSON file. ....36
Figure 18. Find the closest matching question ....36
Figure 19. Retrieving the Answer. ....37
Figure 20. Main function to handle user input and respond ....37
Figure 21. Finding the best match (Question– Answer) in the knowledge base ....38
Figure22. Learning the chatbot ....38
Figure 23. Main Execution. ....39
Figure 24. Chatbot Architecture and Interactions. ....40
Figure 25.Chatbot interface in Arabic. ....41
Figure 26.The Chatbot Interface is in English ....42
Figure 27.The Chatbot Interface is in French ....43

# General Introduction

Artificial Intelligence (AI) increasingly integrates our daily lives with the creation and analysis of intelligent software and hardware, called intelligent agents. Intelligent agents can do a variety of tasks ranging from a labor work to sophisticated operations. A chatbot is a typical example of an AI system and one of the most elementary and widespread examples of intelligent Human-Computer Interaction (HCI) [1].

Now a day's, digital interaction with the devices is a bit limited, with respect to what technology and features being offered. No matter how simple a device technology is, there is always some innovation and new to learn. The chatbot we develop solve this situation by texting with the users and thus solving their problems. Chatbots are the simplest technology that allows to interact with users who are not bots but actual human being... Since chatbots behave like real human beings, Artificial Intelligence (AI) technology is used to develop them. Deep learning is one of the techniques that duplicate the human behavior. It finds similar replies through the dataset and educates itself to reply to the question being asked by users [2].

Chatbots can mimic human conversation and entertain users but they are not built only for this. They are useful in applications such as education, information retrieval, business, e-commerce, and others [3] .

Chatbots became so popular because they have many advantages for users and developers too. Most implementations are platform-independent and instantly available to users without needing installation. Contact to the chatbot is spread through a user's social graph without leaving the messaging application the chatbot lives in, which provides and guarantees the user's identity. Moreover, payment services are integrated into the messaging system and can be used safely and reliably and a notification system re-engages inactive users. Chatbots are integrated with group conversations or shared just like any other contact, while multiple conversations can be carried forward in parallel. Knowledge in the use of one chatbot is easily transferred to the usage of other chatbots, and there are limited data requirements. Communication reliability, fast and uncomplicated development iterations, lack of version fragmentation, and limited design efforts for the interface are some of the advantages for developers too [4]

In light of the rapid technological developments and transformations in methods of communication and service provision, university administrations are looking for new and effective means to improve the student experience and facilitate administration processes. One of these innovative methods is the use of chatbots as a substitute for some human workers . Faculty administration is a complex environment that requires the provision of ongoing services that support students and staff. Modern technology, such as chatbots, comes to facilitate these processes and improve their efficiency. The chatbot is a smart interface that users interact with via text chat. The chatbot guides students and employees and provides support and guidance in real time.

Administrations are also considered one of the sectors that are known to make users very tired. This results from direct communication with customers, which results in many problems, including delays in providing services. You may face a delay in responding to your requests or in providing the required services due to the administrative burden or complex procedures in the case of a large number of students who come to inquire about information that does not require a long time .Students may face the problem of effective communication, that is, a difficulty communicating effectively with departments, whether due to limited communication abilities or other factors such as language or culture. Some students may be exposed to unfair or unjustified decisions by administrations, which can affect their morale and confidence in the system .Things may differ from one university to another or from one institution to another based on several factors, such as institutional culture and specific policies.

Therefore, it seems very important to find an effective solution for interaction between students and administrators working at the faculty to reduce the problems imposed by human interaction. We will explain in the next chapter the aspects in which we can use a chatbot as a substitute for the administrative worker at the faculty.

Deep learning is one of the most promising technology that can solve the problems like vision and the problems encountered in Natural Language Processing (NLP).

We now find ourselves facing a problem: How can we develop Chatbot and make it able to perform well in a specific field using deep learning?

Our manuscript is organized as follows:

In the **General Introduction**, we set the domain for the study, highlighting the significance of chatbots in enhancing university efficiency.

**Chapter 1** explores the potential of chatbots to boost university operations, focusing on the Faculty of Mathematics and Informatics. This chapter delves into the faculty's structure, challenges, and administrative tasks, and discusses the advantages and disadvantages of implementing a chatbot as a faculty administrator.

**Chapter 2** provides a comprehensive overview of chatbots, covering their technology, types, architecture, and the role of Natural Language Processing (NLP). It further examines how deep learning techniques. This chapter also showcases examples of existing chatbots.

**Chapter 3** details the implementation and realization of our chatbot. It outlines the used tools, and explains the methodology for creating the knowledge dataset, finding the best match, and handling user interactions. This chapter also provides insights into the chatbots interface and suggests improvements for user experience and knowledge base expansion.

Finally, the **General Conclusion** summarizes the findings and implications of the study.

# CHAPITRE 01

# Enhancing University Efficiency with Chatbots.

# 1. Introduction

In this chapter, we will talk about how to improve university efficiency by introducing chatbots to faculty services. We took the faculty of Mathematics and Informatics as an example, and we talk about the difficulties that chatbots help to solve and then reduce workers' fatigue. We will also explain things that are done manually and must be known to understand the environment we want to create.

# 2. Boosting University Efficiency with Chatbots

The university, as a central institution of higher education, plays a crucial role in advancing society through education, research, and community engagement. Universities are regarded as centers of knowledge, culture, and progress, providing comprehensive education and fostering critical thinking, problem-solving, and specialized skills in students. They contribute to the innovation through pioneering research that drives scientific, technological, and social advancements, with research findings often influencing public policy and societal improvements. Moreover, universities engage with local and global communities to address pressing issues, promote social justice, and advance sustainability, while also preserving and promoting cultural heritage.

Universities consist of faculties and departments specializing in various fields such as the humanities, social sciences, natural sciences, engineering and technology, medicine sciences, and business. These entities support interdisciplinary research and collaboration to tackle complex issues. Additionally, universities offer administrative and support services including academic affairs, research administration, student services, and financial and operational management. To effectively manage these diverse and complex functions, universities can benefit from innovative technologies that streamline operations and enhance student and faculty experiences.

In this context, tools such as chatbot systems can play a vital role. They can provide instant assistance and information to students, , and faculty staff, improving communication and efficiency across the university. For example, chatbots can assist in answering frequently asked questions, guiding new students, supporting administrative tasks, and even aiding in research inquiries. By integrating such technologies, universities can ensure they remain at the forefront of educational and technological advancement, ultimately contributing to a better and more efficient

academic environment. Through these multifaceted roles, universities significantly contribute to societal development and the creation of a better future for all.

## 3. Faculty of Mathematics and Informatics

The faculty of Mathematics and Informatics is considered as one of the most important academic entities at the university, as it includes a group of departments specialized in various fields related to mathematics and computer science. The faculty aims to provide a distinguished educational and research environment that contributes to the qualification of students and the development of knowledge and technology in society.

### 3.1 Structure (Departments.)

The faculty of Mathematics and Informatics seeks to qualify students and develop their skills in the fields of mathematics and information technology, by providing comprehensive and systematic educational programs that include lectures, practical applications, and scientific research.

The Faculty of Mathematics and Informatics includes two specialized departments:

- **Department of Mathematics**: which specializes in the study of applied mathematics, statistics, data analysis, and operational research.
- **Department of Computer Science:** which focuses on software design, information security, web application development, and information technology, which deals with databases, computer networks, and big data analysis. It also includes artificial intelligence, which revolves around machine learning, natural language processing, and computer vision.

### 3.2 Challenges of the MI Faculty

The faculty administration plays a crucial role in fostering an environment conducive to teaching, research, and scholarly activities while supporting the overall educational goals of the institution.

The faculty administration plays a crucial role in fostering a favorable environment for teaching, research, and scholar activities while supporting the educational goals of the institution..

### 3.3 Main Faculty Administration Tasks

- **Developing Policies and Procedures:** Setting academic policies, developing organizational policies and procedures for the faculty in accordance with its educational, research, and service objectives.
- **Evaluating Academic Programs:** Regularly evaluating and reviewing academic programs of study to ensure that academic standards are met, and developing education curricula.
- **Organizing Examinations and Evaluation:** Organizing examinations and academic evaluation processes fairly and effectively, ensuring adherence to approved standards and procedures.
- **Analyzing Data and Submitting Reports:** Collecting and analyzing academic data and submitting periodic reports to the department and faculty leaderships to make appropriate decisions.
- **Academic Guidance:** Providing academic support and guidance to students to help them achieve their educational and research goals.
- **Evaluating and Developing the Educational Structure:** Evaluating the efficiency and the effectiveness of the faculty's academic structure and developing it to ensure the achievement of the goals of education and research.
- **Periodic Follow-Up and Evaluation:** Reviewing and evaluating the performance of academic policies and procedures and making necessary adjustments to improve performance.
- **Student Guidance and Academic Advising**: Guiding students in choosing appropriate academic paths based on their interests, abilities, and future career goals.
- **Personal and Social Support**: Providing personal and psychological support to students, helping them deal with personal or social challenges, advising on career plans, and academic support.
- **Study Program Planning:** Developing and updating curricula and organizing academic programs to meet student needs and industrial requirements.
- **Academic Recruitment and Training:** Recruiting faculty and administrative staff and providing ongoing training and development to enhance the quality of education and services.
- **Finance and Human Resources:** Managing the faculty's budgets and financial resources, organizing financial operations, managing relationships with funders and external partners, and implementing recruitment and career development policies.

- **Marketing and Public Relations:** Enhancing the faculty's reputation, attracting potential students, and building relationships with important parties, such as media, government institutions, companies, and donors. Organizing events and marketing activities.
- **Evaluation and Quality Assurance:** Monitoring the quality of education and services, evaluating college performance using performance indicators, reports, and periodic reviews.

## 4. How Can We Use AI to Facilitate Administration Tasks?

One of the basic concepts in AI is chatbot. A chatbot is an intelligent program that uses NLP (Natural Language Processing) and can replace a human locator in conversation. The use of a chatbot can perform a variety of tasks in faculty administration, including:

- **Answering Common Inquiries:** Providing immediate and accurate answers to common inquiries about academic schedules, student activities, tuition fees, student accommodation, etc.
- **Academic Guidance and Counseling:** Guiding students regarding different academic paths, graduation requirements, and available subject options.
- **Scheduling Appointments and Meetings:** Facilitating scheduling appointments with academic advisors, teaching assistants, or faculty members, and reminding students of scheduled meetings.
- **Assistance with Registration and Application:** Assisting with registration for courses, submitting applications for academic programs, and submitting required documents.
- **Technical Support:** Providing basic technical support, such as fixing technical problems in student accounts or directing students to available technology resources.
- **Providing Information about Events:** Providing information about university events, lectures, and various student activities, including dates, times, and locations.
- **Basic Academic Advice:** Offering basic advice regarding academic path options, available majors, and graduation requirements.
- **Follow-Up on Requests and Complaints**: Following up on student requests and complaints, providing updates on the status of requests, or directing students to the appropriate channels to resolve issues.

### 4.1. Advantages of a Chatbot as a Faculty Administrator

The use of a chatbot as an alternative to the administration agent in our faculty provides many advantages that contribute to improving work efficiency and saving time and effort, including:

- **Immediate response:** The chatbot allows immediate response to students' and employees' inquiries at any time of the day without the need to wait for official working hours.
- **Cost Savings:** By using a chatbot instead of an admin agent, the university can reduce labor costs and save financial resources.
- **Total availability for full-time:** The chatbot can work 24 hours a day, 7 days a week without stopping, ensuring the availability of technical and administrative support at any time.
- **Presence in multiple places:** A chatbot can be available on multiple platforms such as websites, mobile applications, and text messages, making it available to a wide range of users.
- **Adapting to the size of students:** The chatbot can handle a large number of inquiries simultaneously without any waiting, which makes it suitable for dealing with a large volume of students.
- **Continuous improvement:** Chatbot performance can be continuously developed and improved by updating the knowledge base and training smart models.

By using a chatbot, the faculty administration can enhance the service experience and provide effective support to students and employees in an efficient manner that is available in full-time.

### 4.2. Disadvantages of a Chatbot as a Faculty Administrator

Despite the many advantages when using a chatbot as an alternative to the administration agent in our faculty. Unfortunately, nothing is perfect, thus a chatbot can also have some, disadvantages including:

- **Limitations in human interaction:** Chatbots may lack the ability to effectively handle personal or complex inquiries such as situations that require personal advice or customized solutions.
- **Poor responsiveness to context:** A chatbot may have difficulty understanding the full context of queries, which may lead to it providing inappropriate or inaccurate answers.
- **Information Bias:** A chatbot can be limited by the scope of knowledge it is provided with, which can result in providing inadequate or inaccurate answers in some cases.

- **Lack of human interaction:** Chatbots may lack the ability to provide human interaction that may be important in cases of sensitive or emergency inquiries.
- **Updates and maintenance:** Maintaining the quality performance of the chatbot requires continuous updates and necessary maintenance, which may require additional costs and administrative effort.
- **Limited comprehension ability:** Students and employees may encounter difficulty interacting with the chatbot due to its limited ability to comprehend language and handle complex expressions.
- **Bad connexion of internet:** the chatbot doesn't work if the internet connexion is absent or the signal is very weak, for example when traveling.

Despite these drawbacks, they can be overcome through a balance between the use of technology and human interaction, and continuously developing the chatbot to improve its performance and make the most of its benefits in providing support and services at the university.

## 5. Conclusion

In light of the rapid technological developments and transformations in methods of communication and service provision, university administrations are looking as new and effective means to improve the student experience and facilitate administration processes. One of these innovative methods is the use of chatbots as a substitute for some human workers. In the next chapter, we will talk about chatbots, their types, and their reliance on artificial intelligence and deep learning technology.

# CHAPTER 2

# Chabot's Foundations and Approaches

# 1. Introduction

Artificial Intelligence is changing our daily lives by creating intelligent agents, such as chatbots, that mimic human conversation and are used in various applications such as education, information retrieval, business, and e-commerce. Chatbots are platform independent, real-time, and can be integrated with group conversations or shared like any other contact. It also provides user identity protection, payment services and notification system. Developers benefit from reliable communication, rapid development iterations, and limited interface design effort.

In this chapter, we will talk about some concepts related to chatbots and types of chatbots, and we will mention examples...

# 2. What is a Chatbot?

A chatbot is a software application or web interface designed to mimic human conversation through text or voice interactions, Chatbots are software applications that mimic human conversation, allowing users to interact with digital devices as if they were communicating with a real person. Chatbots use natural language processing (NLP) and machine learning (ML) to analyze user input, understand the intent, and generate appropriate responses. They are becoming more sophisticated, responsive and “natural” in their conversations, as NLP and machine learning (ML) technologies continue to advance. Chatbots are commonly used in customer service to handle inquiries and routine tasks, reducing the need for human agents.

## 2.1. Chatbot Technology

Chatbot technology is an advanced field in artificial intelligence that enables machines to engage in natural and intuitive conversations with humans. This technology relies on key components such as generative AI, deep learning, and natural language processing (NLP). Generative AI creates consistent, contextually relevant, and human-like responses. Deep learning, a subset of machine learning, uses multilayered neural networks to simulate the human brain's structure, allowing chatbots to understand and process vast amounts of data and improve their responses over time. NLP is essential for understanding user inputs, generating appropriate responses, and maintaining the context of conversations. When a user interacts with a chatbot, their input is processed using NLP to understand intent and extract relevant details. The context is analyzed with deep learning models, and the response is generated using generative AI.

## 2.2. Types of Chatbots

Over the last few years, the chatbot field has become so dynamic with the arrival of new Technologies that a precise classification of chatbots has become subjective to the scope of their use. Chatbots could be classified into various categories based on several criteria e.g. mode of interaction, knowledge domain, their usage and the design techniques (response generation method) that are typically employed in building these chatbots. These criteria may include the core design philosophy of the chatbots or the extent to which context needs to be stored and considered in understanding the conversation or the type and purpose of the conversation for which the chatbot needs to be designed [5]

Chatbots can be classified using different parameters: the knowledge domain, the service provided, the goals, the input processing and response generation method, the human-aid, and the build method. Classification based on the knowledge domain considers the knowledge a chatbot can access or the amount of data it is trained upon. Open domain chatbots can talk about general topics and respond appropriately, while closed domain chatbots are focused on a particular knowledge domain and might fail to respond to other questions [6].

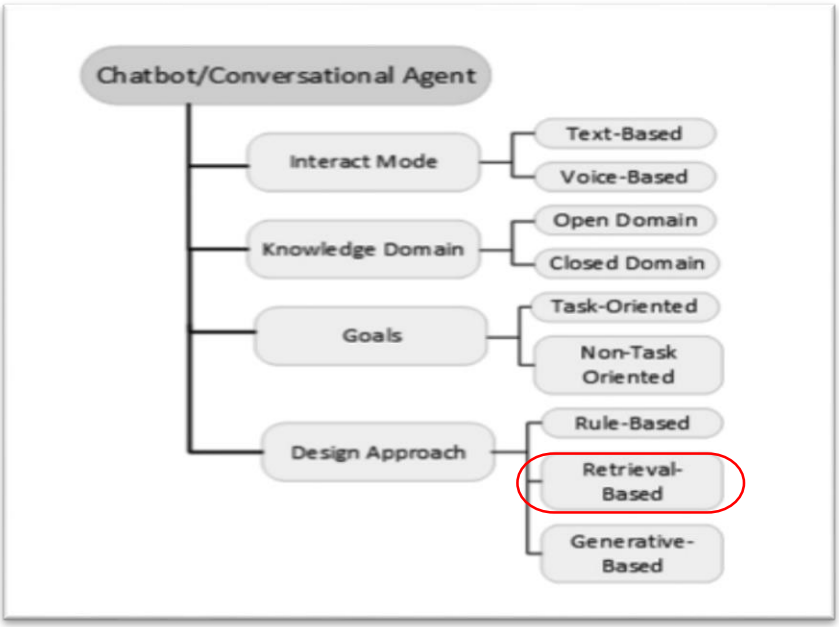


**Figure 1. Classification of Chatbots.**

- **Rule-based**

Model chatbots are the type of architecture which most of the first chatbots have been built with, like numerous online chatbots. They choose the system response based on a fixed predefined set of rules, based on recognizing the lexical form of the input text without creating any new text answers. The knowledge used in the chatbot is humanly hand-coded and is organized and presented with conversational patterns [5].

- **Linguistic Chatbots**

Linguistic chatbots are a type of chatbot that leverages natural language processing (NLP) and linguistic techniques to engage in more natural and contextual conversations with users. These chatbots aim to understand and generate human-like language by analyzing the linguistic structures, semantics, and pragmatics of the user's input.

- **Voice Chatbots**

Voice chatbots, also known as voice assistants or conversational AI assistants, are chatbots that use speech recognition and text-to-speech technologies to enable conversational interfaces through voice input and output. These chatbots are designed to understand and respond to spoken commands and queries, providing a more natural and hands-free interaction experience for users.

- **Dialogue Management chatbot**

Dialogue management involves tracking and maintaining the conversational context, managing dialogue state, and determining the appropriate response or action based on the user's intent and the current dialogue state.

- **Hybrid Model Chatbots**

These bots will be having both features of menu drive chatbots and AI enabled chatbots. These are growing chatbots now-a-days. In real time scenarios may people will be lazy enough to type and answer all the queries asked by the users, So if in another flow, a set of questions with list of options are provided, users can switch the way of answering according to their convenience. Many of the well-developed e-commerce websites will use this hybrid bots for their websites.

They will provide set of products as menu driven and sometimes they will also converse contextually with users to engage them more with their products.

### 2.3. General Chatbot architecture

A general chatbot architecture typically consists of several key components that work together to enable the bot to understand user inputs, process them, and generate appropriate responses.

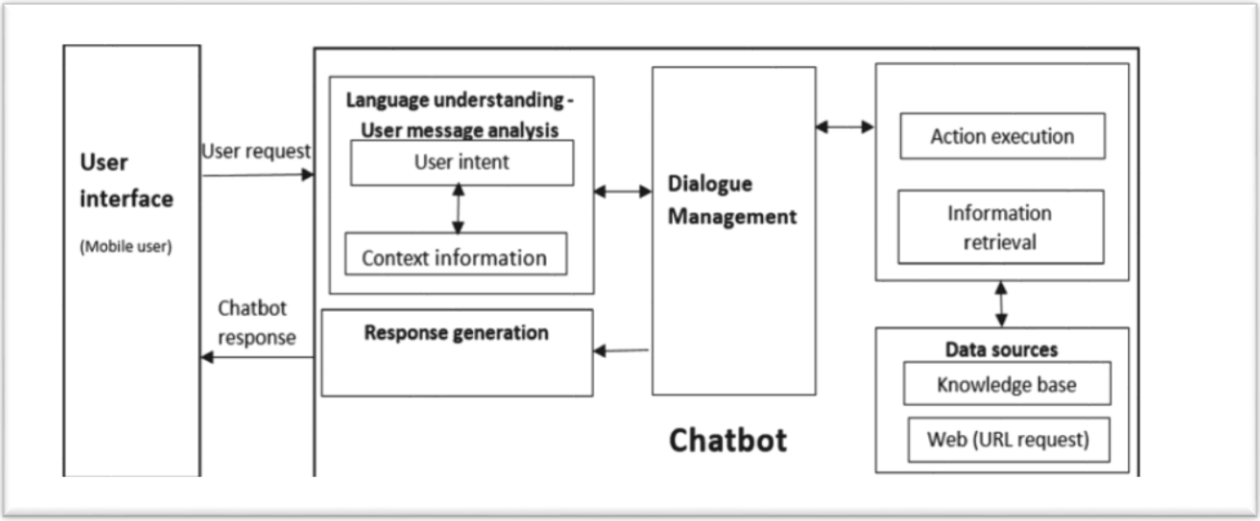


**Figure 2. General Chatbot Architecture.**

## 3. Natural Language Processing (NLP)

### 3.1. Definition

NLP constitutes a core interest in the field of artificial intelligence and computer science. NLP studies comprise theories and methods that enable effective communication between humans and computers in natural language. As a scientific field of study, NLP assimilates computer science, linguistics, and mathematics with a primary goal of translating human (or natural) language into commands that can be executed by computers. NLP consists of two research directions: Natural Language Understanding (NLU) and Natural Language Generation (NLG). The principal mission of NLU is to comprehend the natural language (human language) [7].

By deciphering documents and extracting valuable information for downstream tasks. In contrast, NLG is the production of text in natural languages that are understandable by humans based on the provision of structured data, text, graphics, audio, and video [8].

NLG can be further divided into three categories: text-to-text [9], such as translation and abstract; text-to-other, such as text-generated images [10] ; and other to text (other-to-Text), such as video-generated text [11].

### 3.2. Natural language Processing Techniques and Methods

Natural Language Processing (NLP) techniques and methods encompass a wide range of approaches used to understand, interpret, and generate human language data by computers [12]. Here are some key techniques and methods commonly used in NLP:

- **Encoding**

The first step in NLP is reading data for further analysis. However even in English a recurring challenge is the handling of character encoding. To address this issue, researchers can first write the script in UTF-8. If encoding settings are ignored, loading data directly will fail due to the default encoding for Windows. It is important to make sure every file is in the correct encoding format. Encoding settings can easily be changed in txt files before reading them [13].

- **Cleaning**

Cleaning is an essential preprocessing step in Natural Language Processing (NLP) that involves preparing and transforming raw text data into a format that can be effectively processed and analyzed by NLP algorithms and models. The quality and consistency of the text data heavily influence the performance of NLP systems, making data cleaning a crucial task.

However, cleaning needs may depend on the purpose of the analysis, images, and other nontextual information that must be retained. For example, some researchers have paid explicit attention to emoticons to enrich sentiment dictionaries [13].

- **Noise Removal**

Raw text data often contains various types of noise, such as HTML tags, special characters, URLs, email addresses, and other non-textual elements. Removing these noisy elements is

crucial for improving the quality of the data and preventing potential errors or biases in downstream NLP tasks.

- **Number and Special Character Replacement**: In certain NLP applications, numbers and special characters may not carry significant meaning or may need to be processed differently. These elements can be replaced with placeholders or removed altogether, depending on the specific requirements of the task.

- **Stop Words Removal**: Stop words are common words (e.g., "the," "a," "and," "is") that often carry little semantic value and can be removed from the text to reduce noise and improve the efficiency of NLP algorithms. However, it's important to note that stop word removal should be applied judiciously, as stop words can sometimes be relevant in certain contexts or applications.

- **Tokenization**:

Tokenization is the process of breaking down a sequence of text into smaller units called tokens. These tokens can be words, phrases, symbols, or other meaningful elements. Tokenization is often the first step in many NLP pipelines, as it prepares the text for further processing and analysis. There are many tokenization packages in Python that deliver smart tokenization procedures (NLTK and StanfordCoreNLP) [13].

- **Misspelling:**

The misspelling detection process usually consists of a check on, whether it be via an input string as a valid index or a dictionary word and the correction of spelling errors. N-gram and dictionary lookup are two well-known misspelling detection techniques. For spelling correction algorithms, the most studied are edit distance, similarity keys, rule-based techniques, n-gram-based techniques, probabilistic techniques, and neural networks. Most text-mining packages have prepackaged spellers that can help correct spelling mistakes (e.g., the PyEnchant). In using these spellers, researchers should be aware of the domain-specific language that might not appear in the speller or that the speller might incorrectly “fix.” [13].

- **Part-of-Speech (POS) Tagging:**

POS tagging is an acronym for part-of-speech tagging, a process of attaching each word in a sentence with a suitable tag from a given set of tags (e.g., N represents noun, ADJ represents

adjective, ADV represents verb, etc.) According to the definition, context, and part of speech. An example could be the relationship between adjacent or related words in a phrase, sentence, or paragraph. Some marked corpora can be easily applied in management research, such as NLTK and Pattern in Python. In management research, some researchers have only analyzed nouns and verbs with the help of this technique [14]

- **Stemming and Lemmatization:**

Stemming is the process of reducing a word to its word stem or root form. The stem is the basic part of the word after removing affixes (prefixes and suffixes). For example, the words "playing," "played," and "plays" would all be reduced to the stem "play" after stemming. Lemmatization is a more advanced technique that aims to find the base or dictionary form of a word, known as the lemma. Unlike stemming, lemmatization considers the context and part-of-speech of the word to determine its correct lemma. Several prepackaged stemmers exist in most text-mining tools for example WorldNet, SpaCy and Snowball in Python [13].

### 3.3. NLP Applications

- **Language Modeling**

Language modeling is the task of predicting the probability of a sequence of words or characters in a given language. It is a fundamental component of many NLP applications, including machine translation, speech recognition, and text generation. Language models are often trained using deep learning techniques on large text corpora.

- **Text Summarization**

Text summarization is the task of automatically generating a concise and accurate summary of a longer text document or collection of documents. It involves identifying the most important information and presenting it in a condensed form, while preserving the key points and overall meaning of the original text.

- **Question Answering (QA)**

Question answering systems aim to provide precise answers to questions posed in natural language, by understanding the context and intent of the question and retrieving relevant information from a knowledge base or text corpus.

- **Dialogue Systems and Conversational AI**

Dialogue systems, also known as conversational AI, are designed to engage in natural language conversations with humans. They combine various NLP techniques, such as language understanding, dialogue management, and response generation, to facilitate human-like interactions and assist users with tasks or provide information.

## 4. Combining Deep Learning and NLP

Natural Language Processing (NLP) has witnessed significant advancements in recent years, fueled by the adoption of deep learning techniques. Deep learning models have demonstrated remarkable capabilities in understanding, generating, and processing human language, leading to breakthroughs in various NLP tasks such as machine translation, sentiment analysis, and question answering. The success of deep learning in NLP can be attributed to its ability to learn complex patterns and representations directly from raw text data, without the need for handcrafted features or linguistic rules. Here's an overview of how deep learning is applied to NLP:

### 4.1. Convolutional Neural Networks (CNNs)

Originally developed for computer vision, CNNs have also been successfully applied to NLP tasks by treating text as a sequence of word embedding. CNNs can effectively capture local patterns and extract relevant features from text data, making them useful for tasks like text classification, sentiment analysis, and named entity recognition.

CNNs are designed to learn the spatial features, e.g. edges, corners, textures, or more abstract shapes, that best describe the target class or quantity. The core for learning these features are manifold and successive transformations of the input data (convolutions) on different spatial scales (e.g. via pooling operations). This facilitates identifying and combining both low-level features and high-level concepts. The functioning of a CNN can, hence, be regarded as a mimicry of the animal cortex [15].

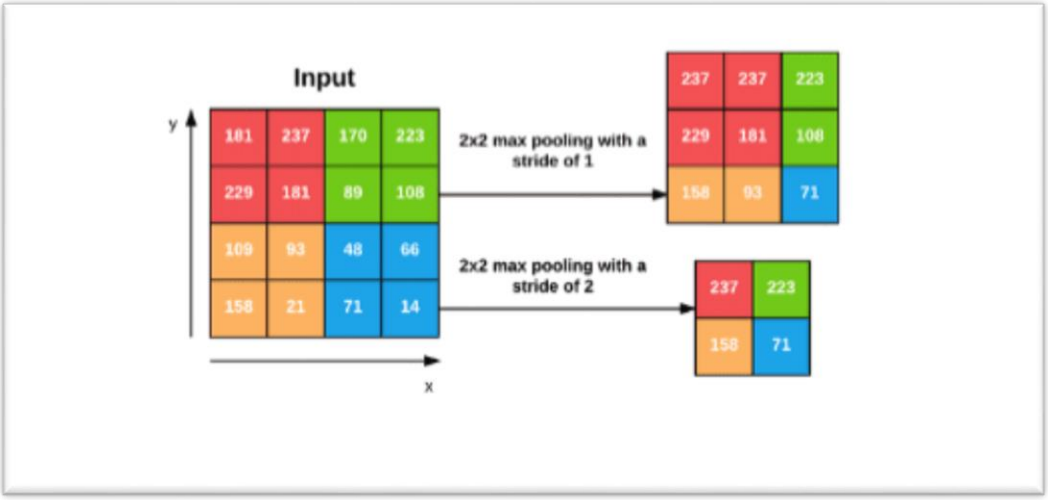


**Figure 3. Convolutional Neural Networks (CNN) and Layer Types.**

### 4.2. Recurrent Neural Networks (RNNs)

RNN, one of the promising DL models, is a suitable learning model for processing sequential data such as speech recognition and language processing. It learns features for time-series data through the memory of previous inputs in the neural network's internal state. Furthermore, RNN can predict future information based on past and present data. However, in the RNN structure, it is difficult to learn stored data for a long time because of the gradient vanishing issue or gradient exploding issue.

Recurrent Neural Networks (RNNs) are a type of neural network architecture which is mainly used to detect patterns in a sequence of data. Such data can be handwriting, genomes, text or numerical time series which are often produced in industry settings (e.g. stock markets or sensors) [16].

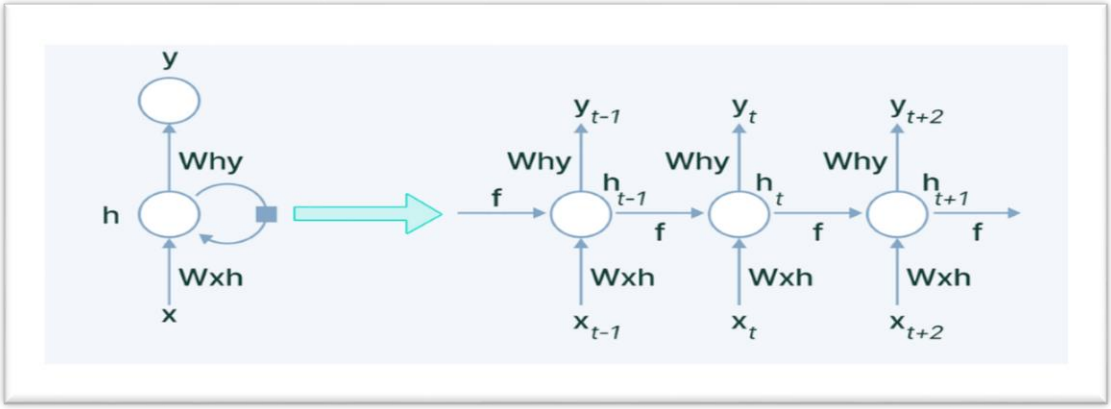


**Figure 4.The Recurrent Neural Networks.**

### 4.3. Long Short-Term Memory LSTM

Long short-term memory (LSTM) is special sort or superior version of a man-made recurrent neural network (RNN) architecture utilized within in the sector of deep learning. LSTM has feedback connections and design to avoid long term dependencies. It can't only process only data points, but also whole sequences of knowledge. For instance, LSTM is applicable to tasks like unsegment data, connected recognition pattern, speech recognition and anomaly detection in network traffic or IDS's (intrusion detection systems). LSTM unit consists four main parts (cell / input gate / output gate/ forget gate) architectures based on Long Short-Term Memory (LSTM), have been designed proving to be capable of solving a variety of complex detection tasks [17].

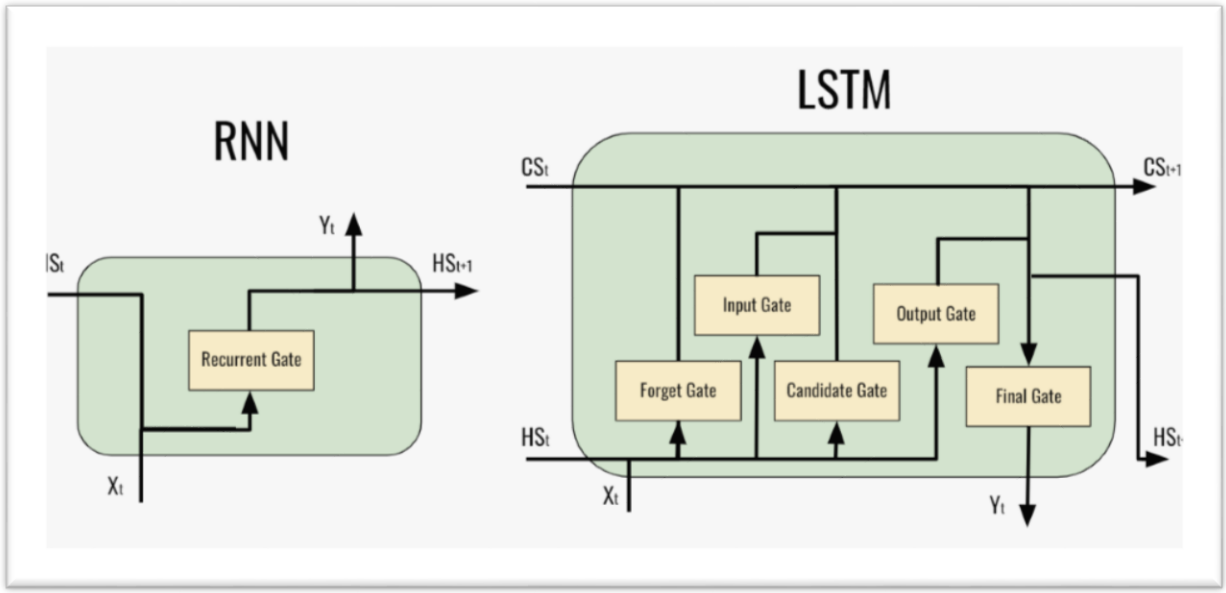


**Figure 5. Long Short-Term Memory LSTM**

### 4.4. Auto Encoder

The auto encoder is a type of neural network that does not require the labeling of data, and therefore it is an unsupervised learning algorithm. The aim is to learn an input function to reconstruct the input to an output of fewer dimensions. It approximates the identity function to get the outcome of a neural network similar to the input. It consists of two main components: an encoder and a decoder:

- **Encoder**: The encoder is responsible for mapping the high-dimensional input data into a lower-dimensional representation, called the "latent space" or "code." This latent representation captures the most important features or patterns in the input data.

- **Decoder**: The decoder takes the low-dimensional latent representation from the encoder and attempts to reconstruct the original high-dimensional input data.

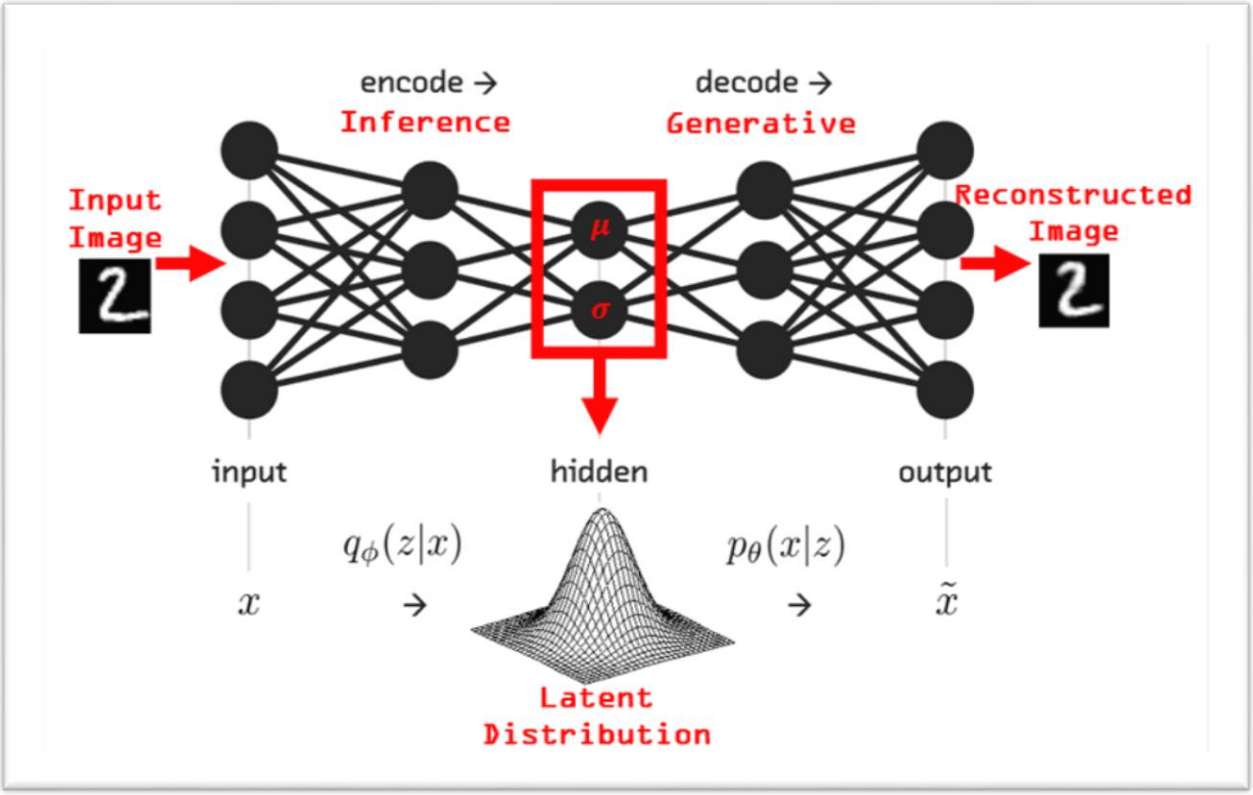


**Figure 6. The Auto Encoder.**

### 4.5. Transformers

Transformers are a relatively recent architecture with remarkable performance on a wide range of NLP tasks. They use a self-attention mechanism to capture global relationships between words in a sentence, allowing them to process input sequences in parallel rather than sequentially. The most well-known transformer model is the transformer-based language model, such as BERT and GPT3, which have achieved state-of-the-art performance on tasks such as text classification, named entity recognition, and language generation.

The Transformer has rapidly become the dominant architecture for natural language processing, surpassing alternative neural models such as convolutional and recurrent neural networks in performance for tasks in both natural language understanding and natural language generation [18].

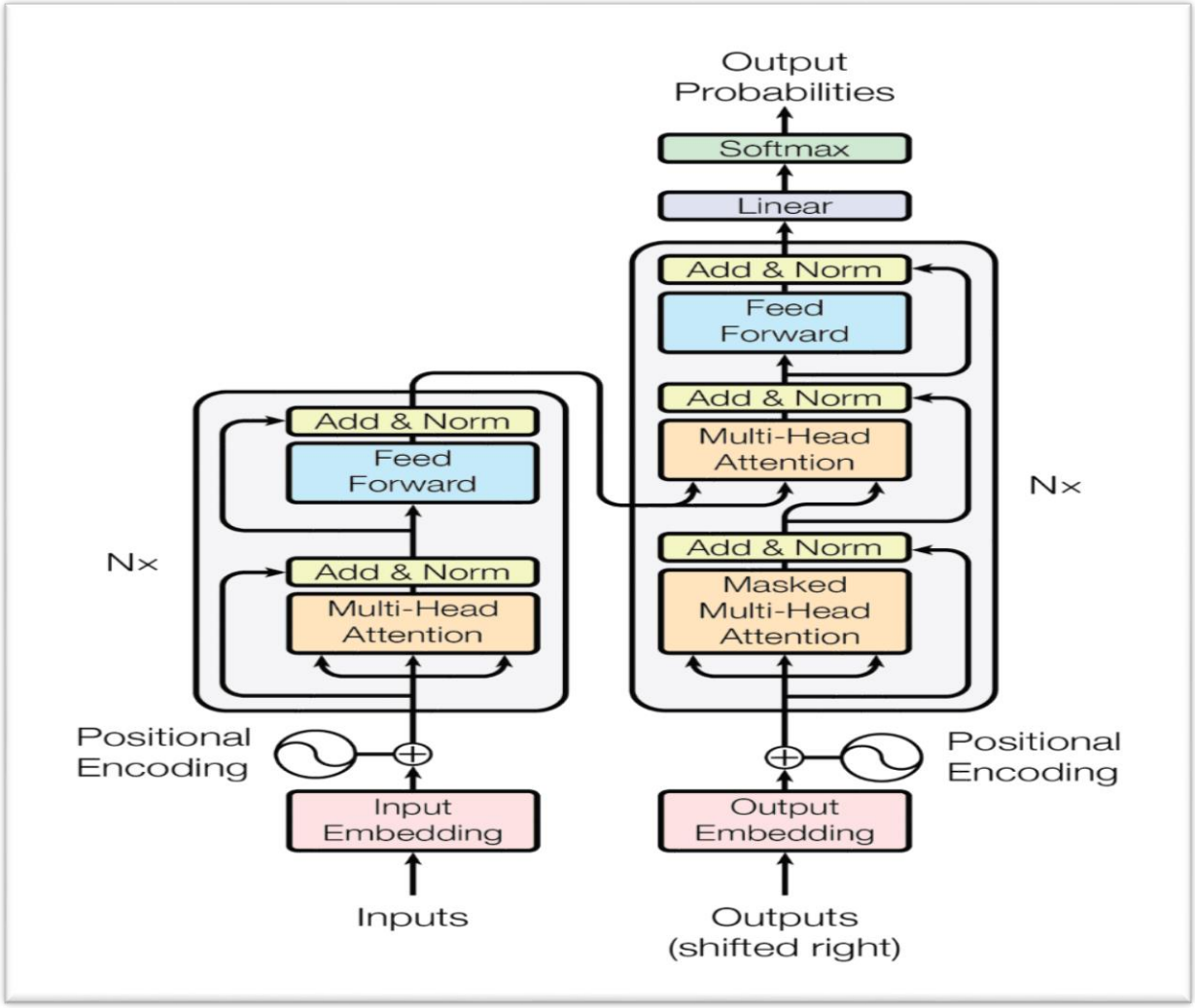


**Figure 7. Transformers**

### 4.6. Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) are an emerging technique for both semi-supervised and unsupervised learning. They achieve this through implicitly modelling high-dimensional distributions of data [19]. They can generate human-like text with improved coherence and diversity compared to traditional language models. They can generate human-like text with improved coherence and diversity compared to traditional language models.

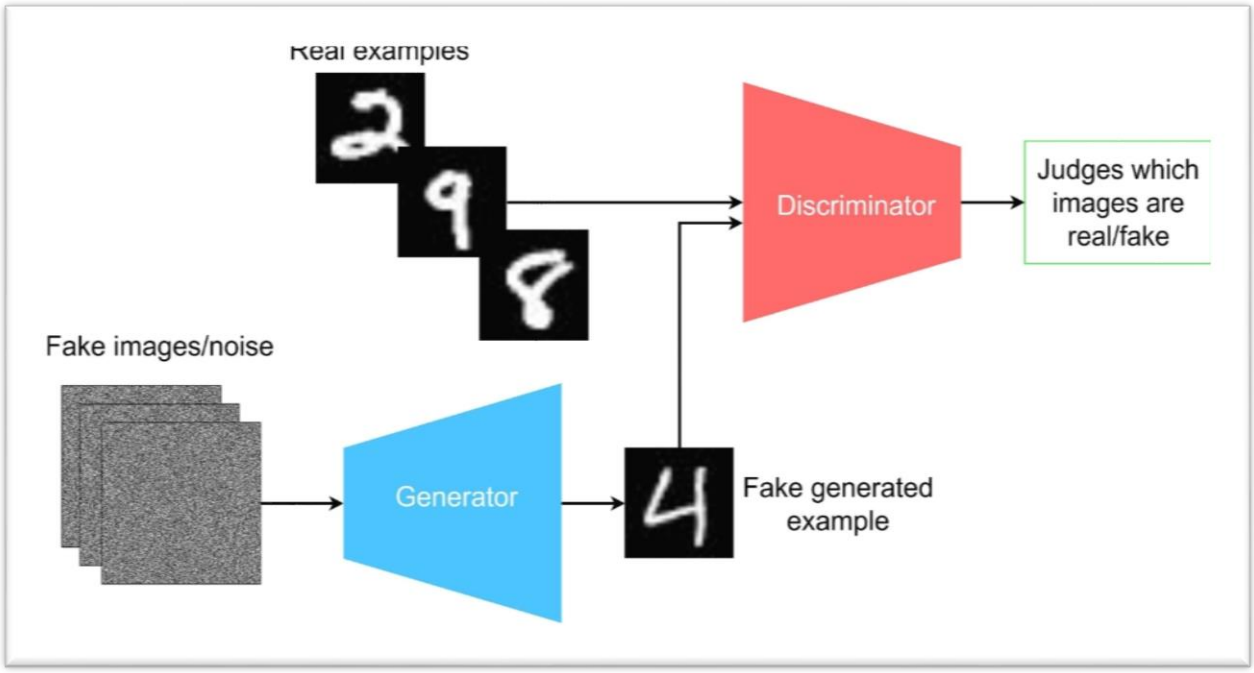


**Figure 8. Generative Adversarial Networks (GANs)**

## 5. Other Models and Techniques to Perform NLP Tasks

### 5.1. Transfer Learning and Pre-trained Language Models

Transfer learning and pre-trained language models have been instrumental in advancing deep learning for NLP. Models like BERT, ArBERT, GPT, and their variants are pre trained on massive text corpora, capturing rich language representations that can be fine-tuned for specific NLP tasks. This approach has significantly reduced the need for large task-specific datasets and has led to improved performance across various NLP applications.

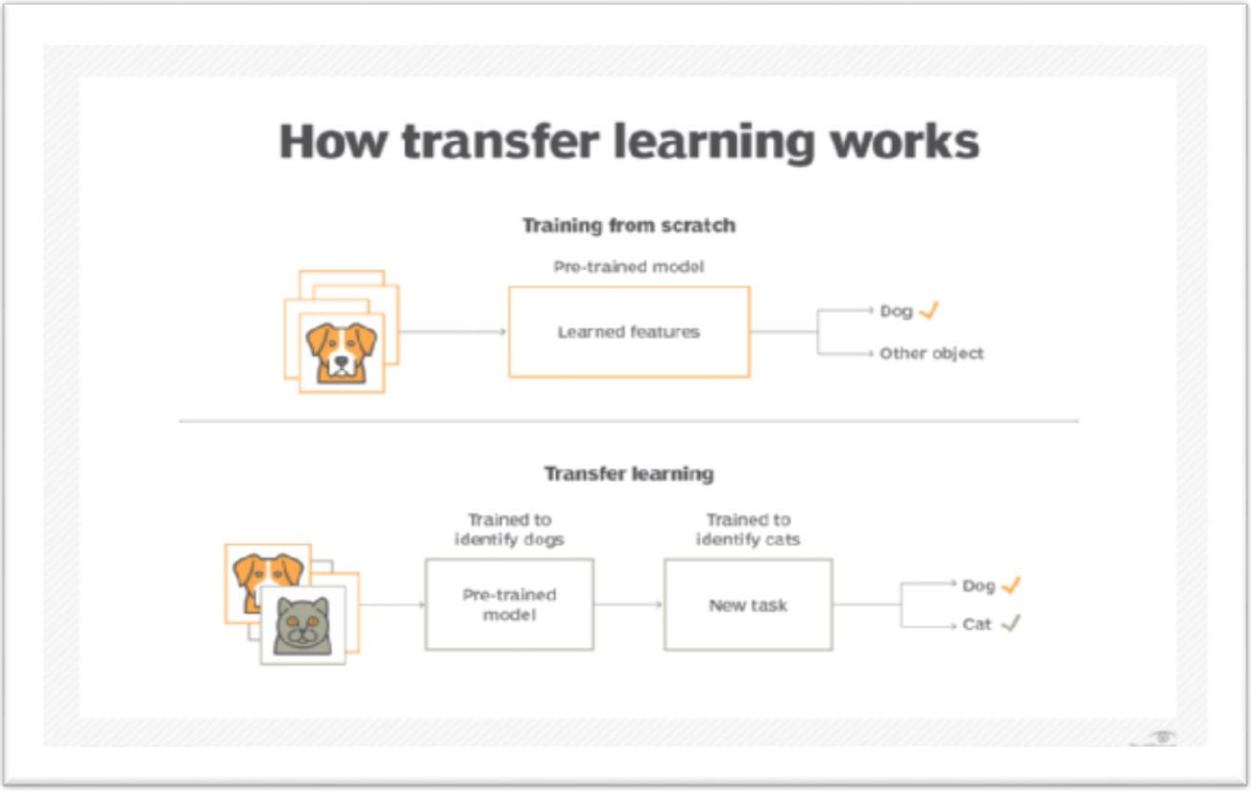


**Figure 9. Transfer Learning and Pre-trained Language Models**

### 5.2. Word Embedding

Word embedding is a fundamental concept in modern natural language processing (NLP) and deep learning. They are dense vector representations of words that capture their semantic and syntactic relationships within a continuous vector space. Word embedding have revolutionized NLP by providing a way to represent words as numerical vectors, which can be efficiently processed by deep learning models. The key idea behind word embedding is that words with similar meanings or contexts tend to have similar vector representations in the embedding space. This property is known as the distributional hypothesis, which states that words appearing in similar contexts tend to have similar meanings.

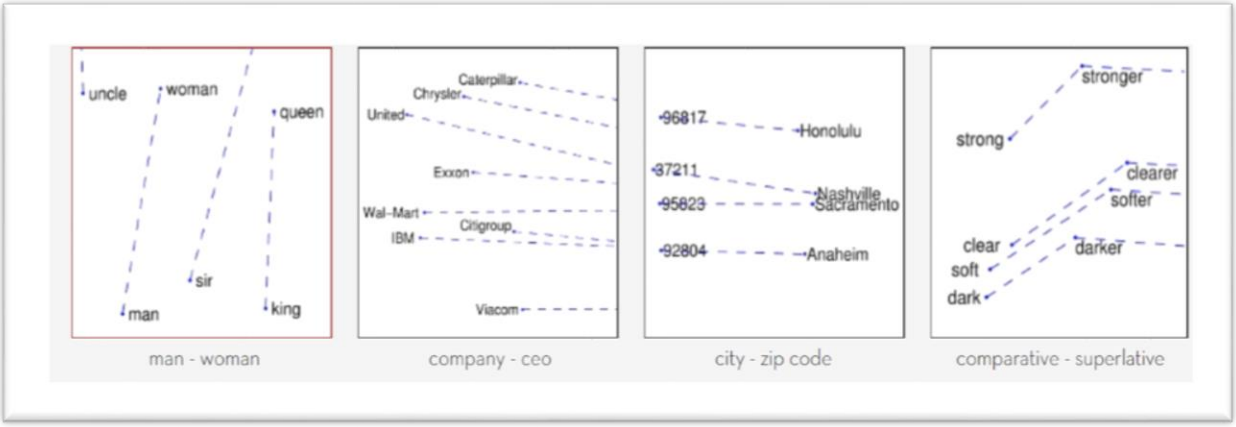


**Figure 11. GloVe.**

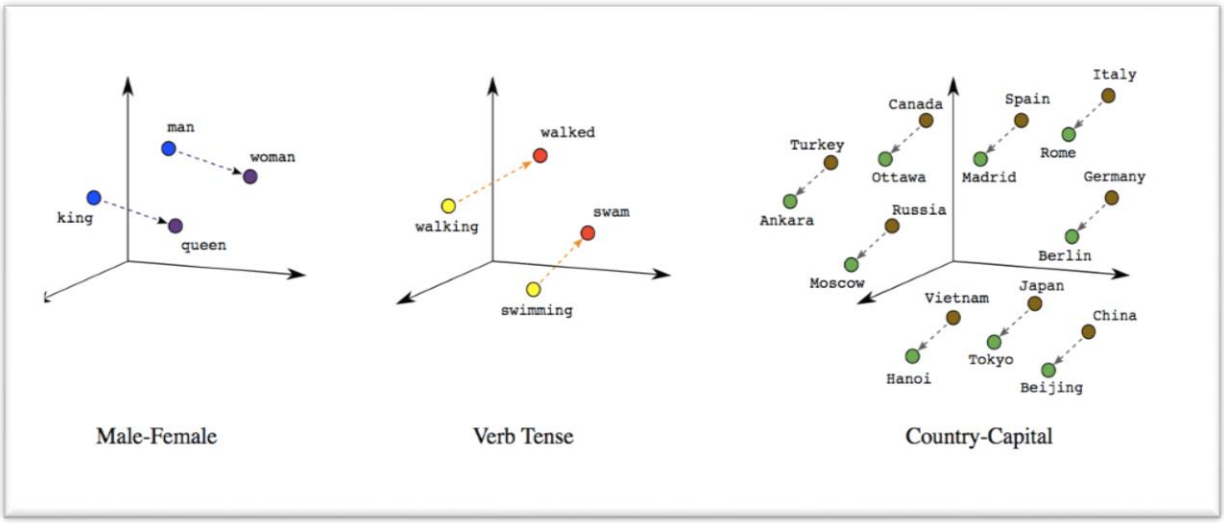


**Figure 10.Word2Vec**

## 6. Examples of Existing Chatbots

### 6.1. SIRI

Siri is a type of chatbot that employs AI and voice-recognition software. Siri is considered a virtual assistant or AI chatbot that can understand natural language through voice and respond accordingly.

Siri uses natural language processing (NLP) and natural language understanding to interpret user inputs and provide relevant responses. Siri is built on artificial intelligence (AI) and machine learning (ML) models that allow it to learn and improve its conversational abilities over time.

The chatbot is powered by large language models and leverages techniques like transformer neural networks to engage in more flexible and contextual conversations.


**Figure 12. Siri.**

### 6.2. ALEXA

Alexa is an artificial intelligence (AI) chatbot that simulates human conversation through voice commands and text chats, providing users with an interactive and automated experience akin to human interaction.

Alexa operates through a combination of machine learning and AI techniques, evolving from traditional rule-based chatbots to more responsive and intelligent systems capable of handling a wide range of user interactions. The Alexa chatbot employs a Seq2Seq model consisting of an Encoder and a Decoder, along with Attention mechanisms, to process and respond to user inputs effectively.


**Figure 13. ALEXA.**

### 6.3. ChatGPT

ChatGPT is an advanced AI conversational agent developed by OpenAI, designed to simulate human conversation through text interactions, answering questions, generating content, and engaging in dialogue similar to human communication.

ChatGPT is a large-scale AI language model developed by OpenAI, based on the GPT-4 architecture, which is an advanced version of the well-known GPT-3.

ChatGPT is part of the Generative Pretrained Transformer series, with "GPT" standing for "Generative Pretrained Transformer," highlighting its ability to generate text, pretrained on a large corpus of Internet text, and utilizing transformer model architecture to understand context.

**Figure 14. ChatGPT**

### 6.4. RASA

RASA chatbots are built using Python programming and leverage natural language processing (NLP) and machine learning techniques to engage in human-like conversations.

RASA utilizes natural language understanding (NLU) models to analyze the text structure and intent of user inputs.

RASA chatbots have a fallback action that is triggered when the model is unable to determine the appropriate response.


**Figure 15. RASA**

## 7. Conclusion

These models, techniques, and studies played a very important role in developing a chatbot for the management of the Faculty of Mathematics and Informatics, which we will talk about in the next chapter.

# CHAPTER 3

# IMPLEMENTATION AND REALIZATION

# 1. Introduction

This chapter is the last part of this thesis, it focusses on the description of the environment, the programming languages, the development tools used in the realization, satge, and an overview of the realized work by giving some screenshots.

# 2. Used tools

## 2.1. Anaconda

The Hub for Data Science and AI Collaboration Expert support and streamlined collaboration at every stage of the data science lifecycle. Source, build, and deploy with ease. Take your work from idea to integration alongside data scientists, open-source contributors, and partners, leveraging leading-edge tools along the way [20].

## 2.2. Python

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [21].

## 2.3.TensorFlow

TensorFlow is an interface for expressing machine learning algorithms and an implementation for executing such algorithms. A computation expressed using TensorFlow can be executed with little or no change on a wide variety of heterogeneous systems, ranging from mobile devices such as phones and tablets up to large-scale distributed systems of hundreds of machines and thousands of computational devices such as GPU cards. The system is flexible and can be used to express a wide variety of algorithms, including training and inference algorithms for deep neural network models, and it has been used for conducting research and for deploying machine learning systems into production across more than a dozen areas of computer science

and other fields, including speech recognition, computer vision, robotics, information retrieval, natural language processing, geographic information extraction, and computational drug discovery [22].

### 2.4. Keras

Keras is an open-source library (MIT license) written in Python which is primarily based on the work done by Google developer François Collet as part of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). The first version of this platform-independent software was published on March 28, 2015. The purpose of this library is to enable neural networks to be rapidly developed [23].

### 2.5. JSON (JavaScript Object Notation)

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language [24].

### 2.6. Spyder

Spyder is a free and open-source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package [25].

### 2.7. UTF8Encoding

UTF8Encoding object to encode a string of Unicode characters and store them in a byte array. The Unicode string includes two characters, Pi (U+03A0) and Sigma (U+03A3), that are outside the ASCII character range. When the encoded byte array is decoded back to a string, the Pi and Sigma characters are still present [26].

### 2.8. Jupyter Notebook

A notebook is a shareable document that combines computer code, plain language descriptions, data, rich visualizations like 3D models, charts, graphs and figures, and interactive controls. A notebook, along with an editor (like JupyterLab), provides a fast interactive environment for prototyping and explaining code, exploring and visualizing data, and sharing ideas with others.

### 2.9. Notepad++

Notepad++ is a free (as in “free speech” and also as in “free beer”) source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GNU General Public License.

Based on the powerful editing component Scintilla, Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment [27].

### 2.10. Numpy

Numpy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more [28].

### 2.11. Vscode (Visual Studio Code)

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET) [29].

### 2.12. NLTK (Natural Language Toolkit)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

### 2.13. Pandas

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language [30].

### 2.14. Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

### 2.15. Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

### 2.16. Scikit-Learn

Scikit-Learn is a free machine learning library for Python. It supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction. The library is built using many libraries you may already be familiar with, such as NumPy and SciPy. It also plays well with other libraries, such as Pandas and Seaborn.

## 3. Methodology Explanation

We used a knowledge-based chatbot algorithm. Knowledge-based chatbots are a type of chatbots that relies on a database or custom knowledge base to save its own set of questions and answers. When a user asks, chatbots match the question with questions in the knowledge base, with the correct answer if a match is found.

Here's a breakdown of the main components and how they work:

### 3.1. Performing the knowledge data set

We have created a JSON file, The JSON format is chosen to store data for several reasons, among these reasons that made us choose this format:

```
19
20    "question": "متى تكون الكلية مفتوحة؟",
21    "answer": "الكلية مفتوحة من الساعة 8 صباحًا حتى 5 مساءً من الاحد إلى الخميس"
22
23
24    "question": "Quand le collège est-il ouvert ?",
25    "answer": "Le collège est ouvert de 8h à 17h du dimanche au jeudi !"
26
27
28    "question": "when is the college open?",
29    "answer": "The college is open from 8am to 5pm from Sunday to Thursday!"
30
```

**Figure 16. Knowledge Base.**

- **Readability and Writ ability:** JSON uses a simple, human-readable format, making it easy to read and write data, containing more than 100 question related to the domain of universes, as well as their answers.

- **Cross-Language Compatibility:** JSON is well-supported in most programming languages, including JavaScript, Python, Java, and others, making it a suitable choice for exchanging data between different components in various applications.

We preferred to use native language for example in our application we have used the three most used languages in our university, including Arabic, French and English. We plan to add other languages as perspective of our work.

### 3.2. Saving the Knowledge Base

We collected frequently asked questions submitted to the college administration by students and placed them in the knowledge base. To update them, the save_knowledge_base () function writes the updated knowledge base to a JSON file (knowledge_base.json).

```
# Save the updated knowledge base to the JSON file
def save_knowledge_base(file_path: str, data: dict):

    with open(file_path, 'w') as file:
        json.dump(data, file, indent=2)
```

**Figure 17. Save the updated knowledge base to the JSON file.**

### 3.3. Finding the Best Match

The find_best_match () function takes the user's input question and a list of questions from the knowledge base. It uses the get_close_matches () function from the difflib module to find the closest matching question in the knowledge base. If a match is found (with a similarity score above 0.6), it returns the matched question; otherwise, it returns None.

```
# Find the closest matching question
def find_best_match(user_question: str, questions: list[str]) -> str | None:
    matches: list = get_close_matches(user_question, questions, n=1, cutoff=0.6)
    return matches[0] if matches else None
```

**Figure 18. Find the closest matching question**

### 3.4. Retrieving the Answer

The get_answer_for_question function takes the matched question and the knowledge base dictionary. It searches for the question in the knowledge base and returns the corresponding answer if found; otherwise, it returns None

```python
def get_answer_for_question(question: str, knowledge_base: dict) -> str | None:

    for q in knowledge_base["questions"]:
        if q["question"] == question:
            return q["answer"]
    return None
```

**Figure 19. Retrieving the Answer.**

### 3.5. Main function to handle user input and respond

Here's an example of the main function in Python:

```python
# Main function to handle user input and respond
def chatbot():

    knowledge_base: dict = load_knowledge_base('knowledge_base.json')

    while True:
        user_input: str = input("You: ")

        if user_input.lower() == 'quit':
            break
```

**Figure 20 . Main function to handle user input and respond**

### 3.6. Learning the Chatbot to Answer a New Question

If no match is found, the chatbot asks the user to learn him how it answers to the question. If the user provides a new answer, it is added to the knowledge base, and the updated knowledge base is saved using save_knowledge_base().

```
# Finds the best match, otherwise returns None
best_match: str | None = find_best_match(user_input, [q["question"] for q in knowledge_base["questions"]])

if best_match:
    # If there is a best match, return the answer from the knowledge base
    answer: str = get_answer_for_question(best_match, knowledge_base)
    print(f"Bot: {answer}")
else:
    print("Bot: I don't know the answer. Can you teach me?")
    new_answer: str = input("Type the answer or 'skip' to skip: ")

    if new_answer.lower() != 'skip':
        knowledge_base["questions"].append({"question": user_input, "answer": new_answer})
        save_knowledge_base('knowledge_base.json', knowledge_base)
        print("Bot: Thank you! I've learned something new.")
```

**Figure 21. Finding the best match (Question – Answer) in the knowledge base**

```
You: do you have a name?
Bot: I don't know the answer. Can you teach me?

Type the answer or 'skip' to skip: I am a Chatbot.
```

**Figure22. Learning the chatbot**

### 3.7. Main Execution

The last part of the code checks if the script is being run as the main program (if __name__ == "__main__":) and calls the chatbot function if that's the case.

```
if __name__ == "__main__":
    chatbot()
```

**Figure 23. Main Execution.**

## 4. Chatbot Interaction

The chatbot function serves as the central component of the program, orchestrating its functionality. Initially, it initializes by loading the knowledge base through the load_knowledge_base () function, ensuring it has access to the necessary information for interaction. Upon initialization, the chatbot enters into a continuous loop, awaiting user input. This loop remains active until the user decides to terminate the interaction by typing 'quit', prompting the program to gracefully conclude.

Within each iteration of the loop, the chatbot function endeavors to enhance user experience by identifying the most relevant response to the input provided. It achieves this by employing the find_best_match () function, which searches through the knowledge base to pinpoint the closest match to the user's query. Upon locating a suitable match, the corresponding answer is retrieved from the knowledge base utilizing the get_answer_for_question () function. This retrieved response is then displayed to the user, enriching the conversation.

However, in cases where the chatbot fails to identify a suitable response from its existing knowledge base, it adopts a proactive approach. It prompts the user to contribute to its learning process by teaching it the answer to the queried question. Upon receiving the user's input, containing the sought-after answer, the chatbot seamlessly integrates this newfound knowledge into its existing repository. It accomplishes this by adding the new answer to the knowledge base and subsequently saving the updated knowledge base using the save_knowledge_base() function, ensuring that future interactions benefit from the expanded knowledge base. Through this

iterative process of interaction and learning, the chatbot continually evolves, providing more accurate and comprehensive responses to user inquiries over time.

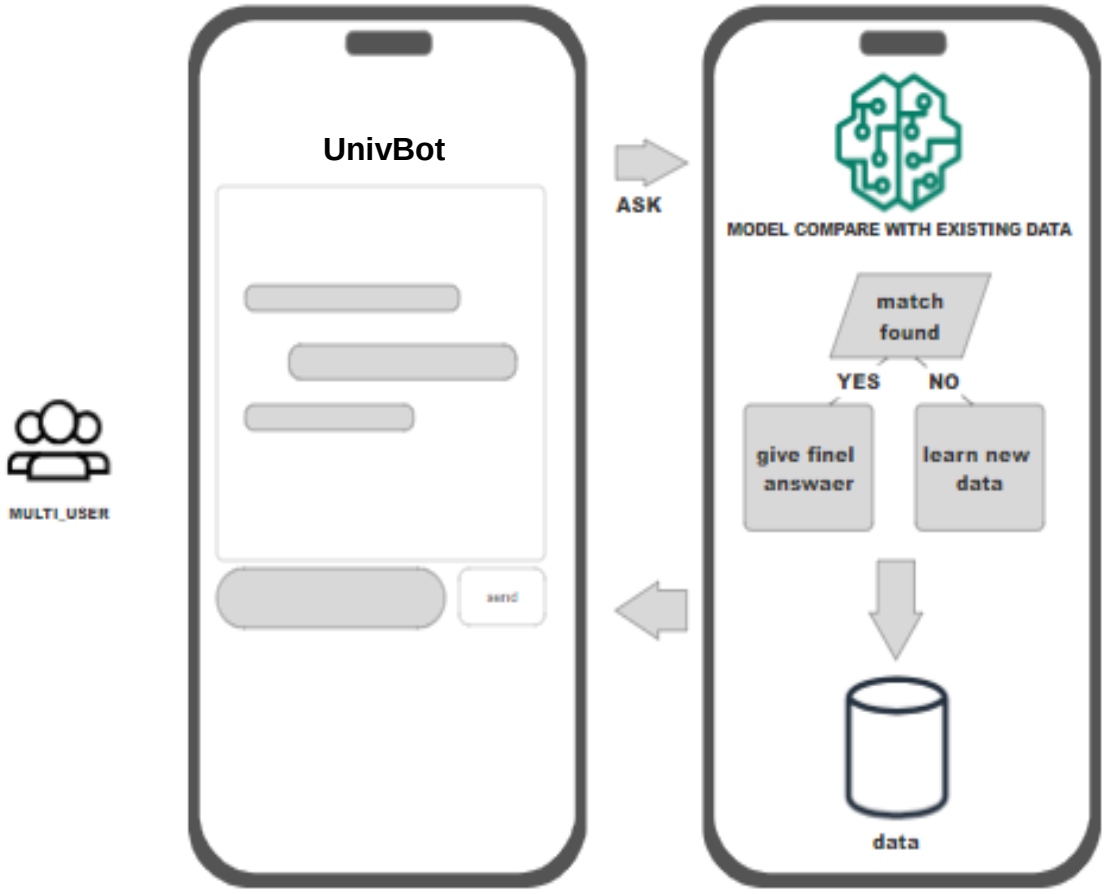


**Figure 21. Chatbot Architecture and Interactions.**

## 5. Explanation of Interfaces :

The chatbot is able to handle various inquiries, such as changing specializations, knowing the dates of exhibitions and events, and inquiring about available specializations.

### 5.1. Interactivity:

The interaction is guaranted using an official natural language (Arabic, English, French) and the chatbot appears to be programmed to answer a wide range of common questions.

The chatbot provides clear directions, such as submitting a formal application to the responsible of Admissions and registration or visiting a specific website for additional information.

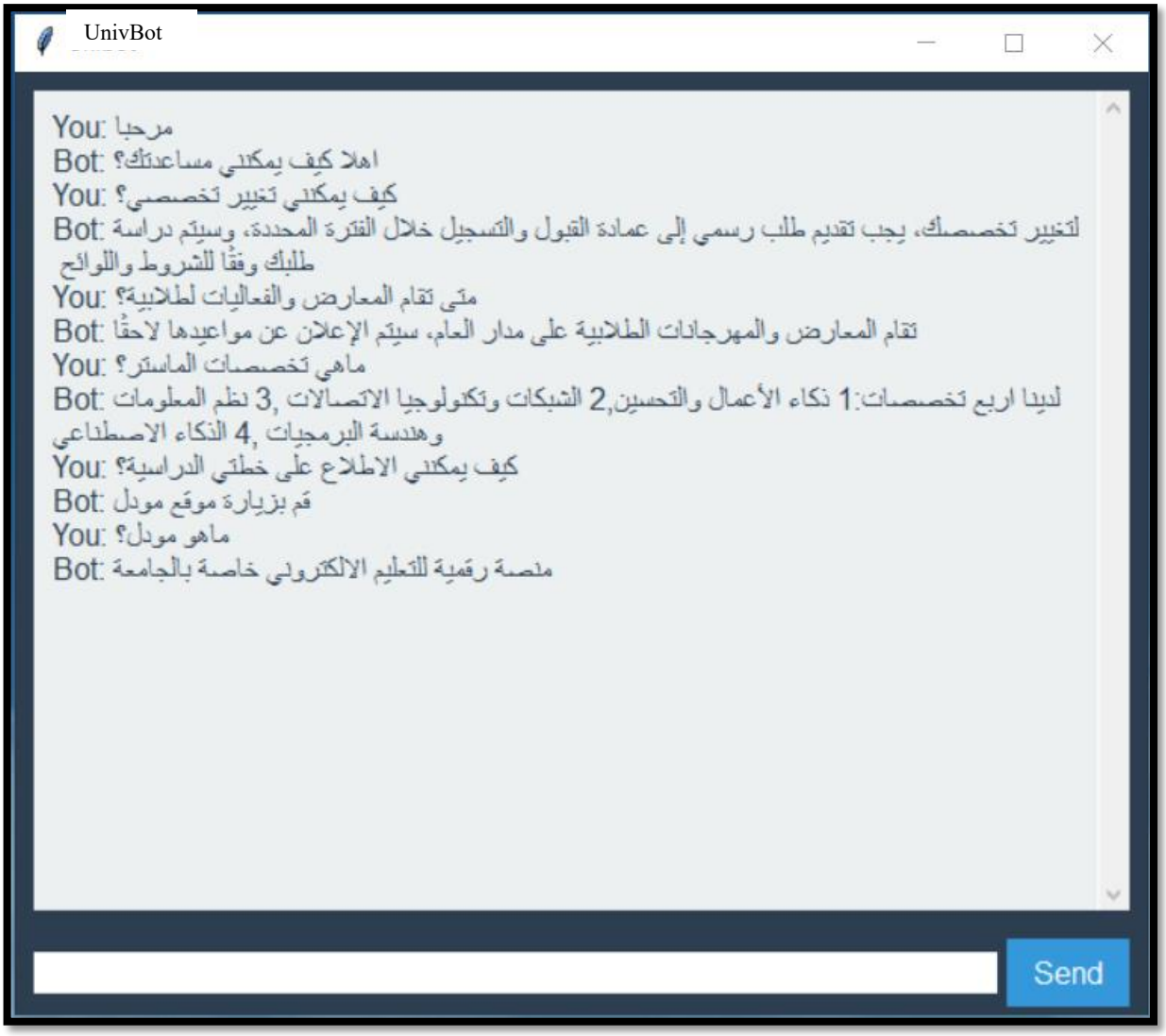


**Figure 22. Chatbot interface in Arabic.**

### 5.2. Using other Languages

The language used is formal and professional, reflecting an academic environment.The robot uses clear and direct language, which makes it easy for users to understand instructions and guidelines. We have used the three languages: Arabic, English and French.

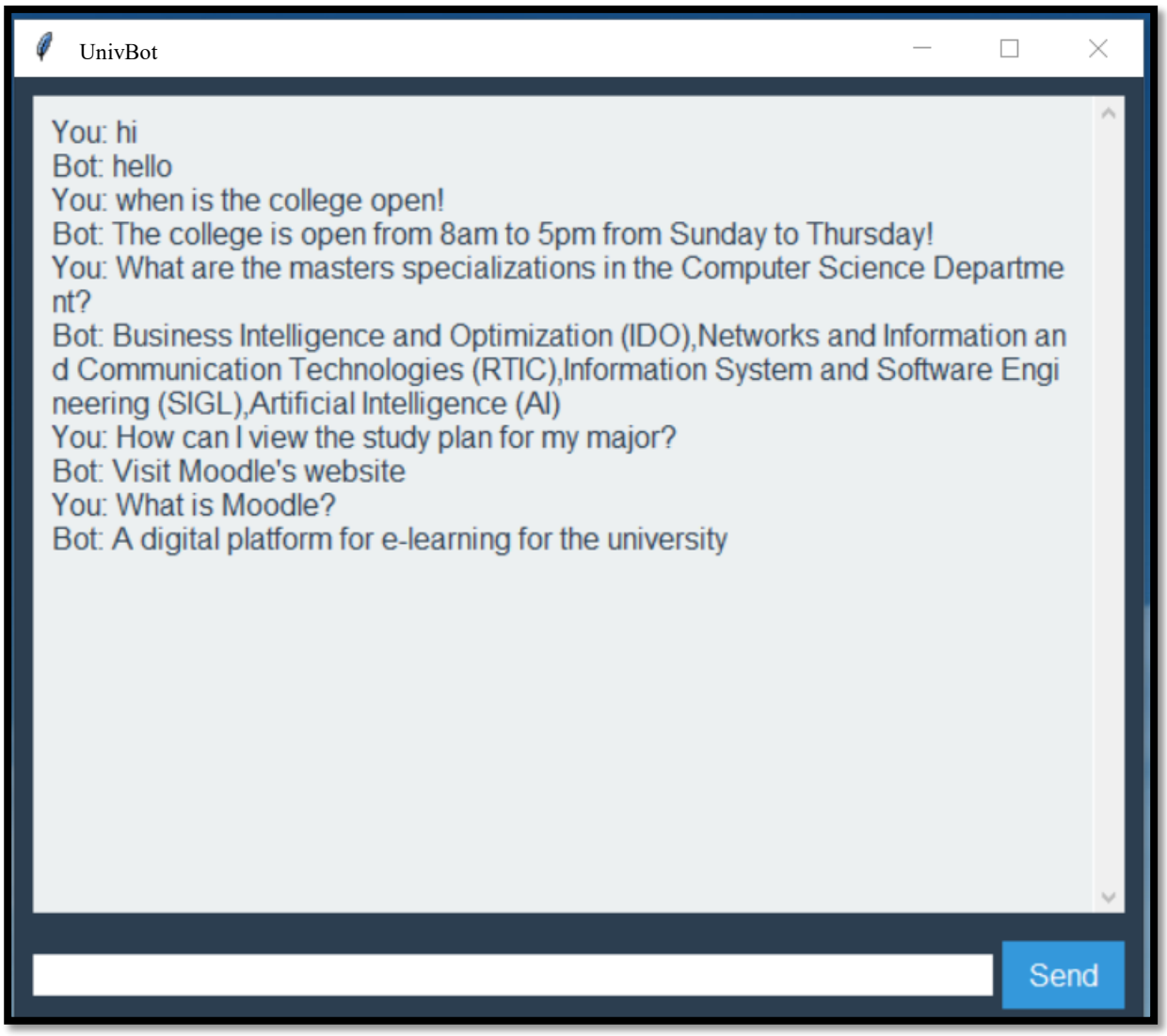


**Figure 23.The Chatbot Interface is in English**

### 5.3. Comprehensiveness:

The robot covers multiple areas of academic inquiries, including student activities, study plans, and different majors.

It allows users to know details about the academic system and available electronic resources, such as the Moodle platform.

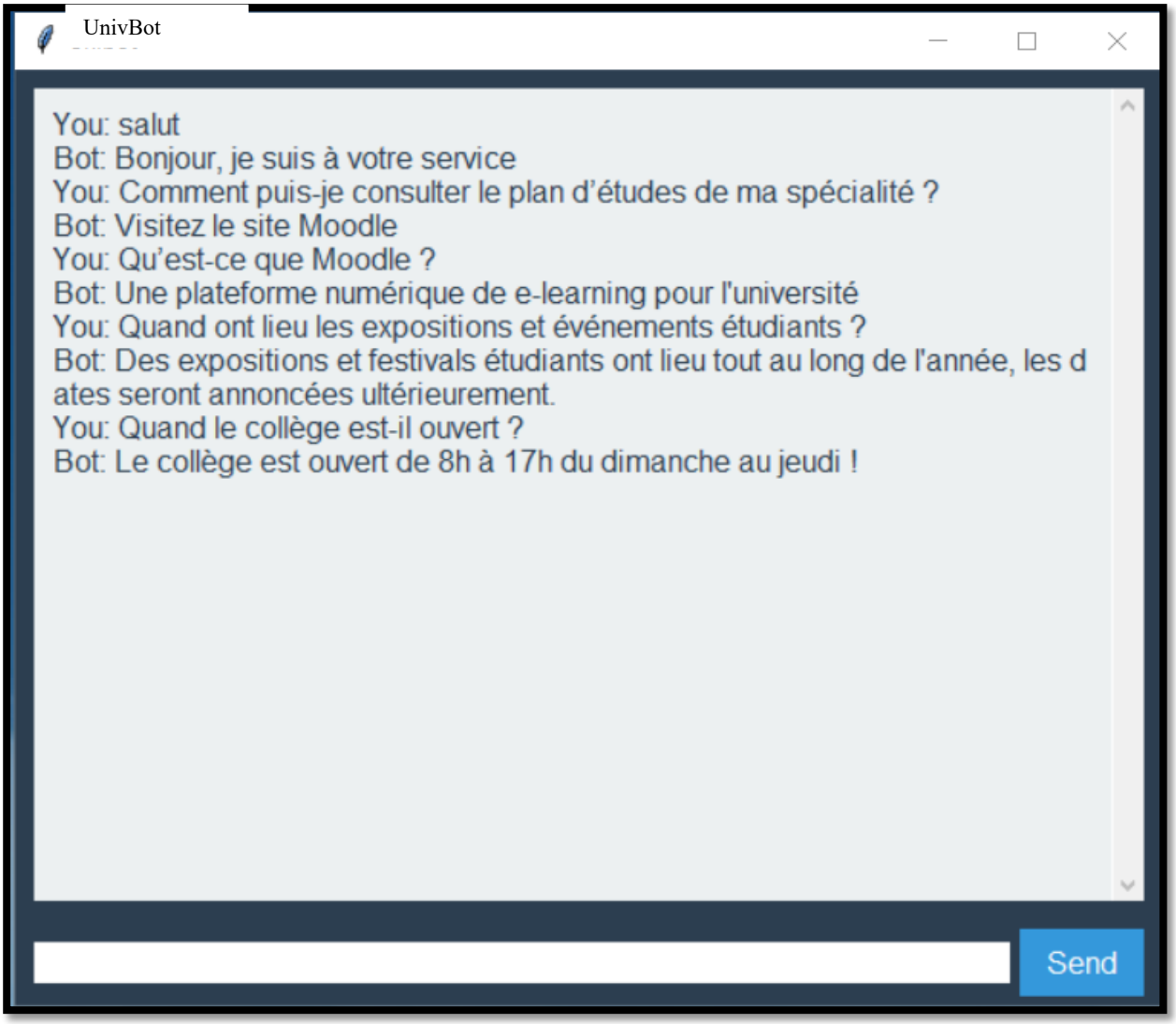


**Figure 24.The chatbot interface is in French**

## 6. Suggestions for Improvement

UnivBot provides an efficient and simple interactive experience for users, which contributes to improving academic skills and reducing the hardness of academic staff. With some improvements, this tool can become more effective and attractive to users.

### 6.1. User interface

Adding visual elements such as icons or images can make the interface more attractive and easier to use.

### 6.2. Expanding the knowledge base

Update the robot database periodically to ensure the provision of the latest information and answers to new inquiries.

### 6.3. Improve the interaction

Developing a learning feature from previous conversations can help the chatbot to provide more accurate and personalized answers to users.

### 6.4. Create an interactive version on smartphone

Creating an interactive version of UnivBot on smartphones enhances accessibility for students and faculty, allowing them to obtain information and services easily, anytime, and anywhere. This chatbot enables instant interaction, providing a personalized experience that effectively addresses academic and administrative inquiries. Additionally, it facilitates quick updates and broad distribution, ensuring that everyone receives the most current information.

### 6.5. Voice Recognition and Voice Answering Technology

Voice recognition and voice answering technology can be integrated to make interacting with the chatbot more natural and seamless, meeting the needs of users who prefer voice interaction over text. By merging voice technologies, users can converse naturally with the chatbot, ask questions, and receive answers without the need for typing, providing them with a more efficient and engaging interactive experience.

### 6.6. Adding Other Languages

UnivBot communicates with users using three (03) basic languages, notably: Arabic, English, French. One of the possible improvements is to add other languages to have more universal chatbot.

### 6.7. Adding Local Dialects

Sometimes users cannot interact with the chatbot using a standard language (Ar, En, Fr) because of their weak level in this language or because the locator is another person not the interested student (his father, his mother, colleague, other person who doesn't speak the three languages). In this case, an efficient solution is to introduce some dialects to facilitate communication with the chatbot (Algerian Dialect like Chaoui, Tamazight, Mezabit, ...).

### 6.7. Adding UnivBot to the Official Website of the University

One of our objectives after improving chatbot functionalities is to integrate it in the official website of the university which allows to a large number of visitors to use it directly.

## 7. Kinds of Question/Answers Processed by UnivBot

Univer-bot processes a variety of questions, primarily focused on information related to a faculty or university setting. These questions can be categorized into the following categories:

- Greetings and Basic Interactions.
- Operational Hours.
- Academic Programs
- Specialties
- Events and Activities
- E-learning Platforms
- Studies Planning
- Points sheets
- Calendar of regular session exams
- Calendar of remedial course exams
- Debt students
- Teachers' timing

- Fraud penalties
- Moving between groups
- Good conduct certificate
- Professors-Students communication
- Conditions of employment
- Academic holidays
- Diplomas
- Students' Evaluation
- Justifying absence
- Absences management

These types of questions cover essential aspects of college and university life, ensuring that UnivBot can assist students, prospective applicants, and other stakeholders effectively.

## 8. Conclusion

In this chapter, we highlighted all the necessary techniques to implement our own chatbot, which will be used by students of the faculty of Mathematics and Informatics. We have used the appropriate environment and all useful and necessary tools to create the chatbot, as well as to provide database examples, without forgetting interfaces that facilitate communication Bot-User using the most used languages in Algeria, namely: Arabic, English, and French.

# General Conclusion

In light of rapid technological developments and radical changes in methods of communication and service delivery, university administrations are constantly seeking to adopt new and effective means to improve the student experiences and facilitate administrative processes. Among these innovative methods, we find the use of chatbots as an assistant to human that work at the university.

The use of chatbots in the university administration includes providing immediate responses to student and employee inquiries at any time of the day, reducing the need to wait until official working time. In addition, the chatbot contributes to reducing costs by reducing the need for human employees, and provides technical and administrative support at any time, which enhances the effectiveness and the speed of services.

Despite the many benefits, chatbots face some challenges. Among these challenges we can note the following: the limited ability to handle personal or complex inquiries, the difficulty of understanding the full context of inquiries, information bias due to the scope of knowledge available to the chatbot, and the necessity of constantly updating and maintaining the chatbot to ensure high-quality performance, which may require additional administrative costs and efforts.

The chatbot can be developed continuously by updating the knowledge base and training smart models like Large Language Models LLMS. This requires adding real-time data and information periodically, to ensure that the chatbot is able to provide accurate and real-time answers to all received inquiries. In addition, machine learning techniques can be used to improve a Chabot's abilities to understand context and answer complex questions more efficiently.

Enhancing the interaction between technology and humans is a crucial step to achieve a better balance between automation and human interaction. A system could be developed whereby

a chatbot can handle simple and repetitive inquiries, while more complex inquiries are directed to specialist university staff. This approach ensures high-quality service to users and reduces the hardness of human staff, enabling them to focus on tasks that require human intervention.
In addition, introducing visual elements and voice interaction options could be considered to make the user experience more engaging and user-friendly. For example, an interactive user interface could be developed containing educational videos or infographics that visually explain the required information. Also, voice recognition and voice answering technology can be integrated to make interacting with the chatbot more natural and seamless, meeting the needs of users who prefer voice interaction over text.

Expanding the idea of using a chatbot could include introducing it into other administrative departments, such as the grants, transportation, and accommodation departments. In the grants section, the chatbot can provide information about requirements, deadlines, and application procedures. In the transportation section, it can provide transportation schedules and available services. In the Accommodation Department, it can help students find suitable accommodation and answer inquiries related to accommodation.

Moreover, chatbots can be used for some academic standards and create specialized scientific chatbots. These robots can provide immediate assistance to students with homework assignments, provide additional explanations, or provide reviews of course concepts. This type of chatbots can act as a teaching assistant, enhancing the learning experience and allowing students to make the most of their time.

The present project can be considered as the first step in a famous project scheduled by the MHESR (Ministry of Higher Education and Scientific Research) which places the Algerian university in the fourth generation.

Furthermore, working to continuously develop the chatbot and expanding its use to include multiple administrative and academic fields will enhance the university's efficiency and make it able to provide better and more comprehensive services to its students and employees. This technical development will not only contribute to improving the user experience, but will also enable the university to achieve high and effective performance in all its aspects.

## Bibliography:


[1] R. K. H. Bansal, «human computer interaction,» vol. 8, p. 53, 2018.

[2] A. D. Neeraj Singh Kadayan, «CHATBOT USING DEEP LEARNING,» p. 1, April 2019.

[3] E. A. B.A. Abu Shawar, «Chatbots: are they really useful?,» pp. 29-49, 2007.

[4] A. Bogliolo, L. Klopfenstein, S. Delpriori et S. Malatini, «The rise of bots: a survey of conversational interfaces,patterns, and paradigms,» pp. 555-565, 2017.

[5] K. Ramesh, S. Ravishankaran, A. Joshi et K. Chandrasekaran, A Survey of Design Techniques for Conversational Agents, vol. 750, S. G. D. K. L. C. D. (. Kaushik, Éd., Springer, Singapore, 2017, p. 336–350.

[6] K. Nimavat et T. Champaneria, «Chatbots: an overview types, architecture, tools and future possibilities,» *International Journal of Scientific Research and Development (IJSRD),* vol. 5, p. 1019–1024, 2017.

[7] R. C. Schank, « Conceptual dependency: A theory of natural language understanding,» *Cognitive Psychology,* vol. 3, n° %14, pp. 552-631, 1972.

[8] D. D. McDonald, Handbook of Natural Language Processing, vol. 2, 2010, pp. 121-144.

[9] Genest, P.-E., Lapalme et G., «Framework for abstractive summarization using text-to-text generation,» chez *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, 2011.

[10] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang et X. He, «Attngan: Fine-grained text to image generation with attentional generative adversarial networks,» chez *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] M. Rohrbach, W. Qiu, I. Titov, S. Thater et M. &. S. B. Pinkal, «Translating video content to natural language,» chez *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[12] D. Lee, K. Hosanagar et H. S. & Nair, «Advertising content and consumer engagement on social media: Evidence from Facebook,» *Management Science,* vol. 64, n° %111, pp. 5105-5131, 2018.

[13] Y. Kang, Z. Cai, C.-W. Tan, Huang, L. Qian et Hefu, «Natural Language Processing (NLP) in Management Research,» p. 26, 2020.

[14] Y. Pan, P. Huang et A. & Gopal, «Storm clouds on the horizon? New entry threats and R&D

investments in the U.S. IT industry,» *Information Systems Research,* vol. 30, n° %12, pp. 540-562, 2019.

[15] C. Angermueller, T. Parnamaa, L. Parts et O. Stegle, «Deep learning for computational biology,» *Molecular Systems Biology,* 2016.

[16] J. Chung, Caglar Gulcehre, Kyunghyun Cho et Y. Bengio, «Gated Feedback Recurrent Neural Networks,» chez *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, Lille, France, 2015.

[17] G. Loganathan, J. Samarabandu et X. Wang, «Sequence to Sequence Pattern Learning Algorithm for Real-Time Anomaly Detection in Network Traffic.,» chez *IEEE Canadian Conference on Electrical & Computer Engineering (CCECE).*, 2018..

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser et I. Polosukhin., «Attention is All You Need,» chez *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[19] I. Goodfellow, J. Pouget-Abadie, M. M. B. Xu, D. Warde-Farley, S. Ozair, A. C. et Y. Bengio., «Generative Adversarial Nets.,» chez *Advances in Neural Information Processing Systems (NIPS 2014)*, 2014.

[20] «Anaconda Hub,» [En ligne]. Available: https://www.anaconda.com/products. [Accès le 30 05 2024].

[21] «What is Python? Executive Summary,» 2001. [En ligne]. Available: https://www.python.org/doc/essays/blurb/. [Accès le 30 05 2024].

[22] «Citing TensorFlow,» [En ligne]. Available: https://www.tensorflow.org/about/bib. [Accès le 30 05 2024].

[23] «keras,» [En ligne]. Available: https://keras.io/. [Accès le 30 05 2024].

[24] «Introducing JSON,» JSON, [En ligne]. Available: https://www.json.org/json-en.html. [Accès le 30 05 2024].

[25] «Overview,» [En ligne]. Available: https://www.spyder-ide.org/. [Accès le 30 05 2024].

[26] «UTF8Encoding Class,» [En ligne]. Available: https://learn.microsoft.com/en-us/dotnet/api/system.text.utf8encoding?view=net-8.0. [Accès le 05 30 2024].

[27] «What is Notepad++,» [En ligne]. Available: https://notepad-plus-plus.org/. [Accès le 30 05 2024].

[28] «NumPy,» [En ligne]. Available: https://numpy.org/. [Accès le 05 30 2024].

[29] «Getting Started,» [En ligne]. Available: https://code.visualstudio.com/. [Accès le 30 05 2024].

[30] «pandas,» [En ligne]. Available: https://pandas.pydata.org/. [Accès le 15 MAY 2024].

## ABSTRACT

In light of rapid technological developments, the university administrations are seeking to adopt new and sophisticated means to improve the student experience and facilitate administrative processes. One of these means is to use chatbots as an assistant to university students and employees. Chatbots allow immediate responses to student inquiries and provide technical and administrative support in full-time, which reduces costs and improves the student experience. Based on artificial intelligence technologies and natural language processing techniques, we have developed our own chatbot called UnivBot that permits to answer to a large range of students' questions and reduces the large volume of work ensured by the dministrative staff so that they can devote themselves to other tasks. Among the characteristics of UnivBot that it is able to be improved constantly and quickly by updating the its knowledge base, and training language models and deep learning techniques to provide answers in real-time and with a high accuracy.

**الملخص :**

في ظل التطورات التكنولوجية السريعة، تسعى إدارات الجامعات لاعتماد وسائل حديثة ومتطورة لتحسين تجربة الطلاب وتسهيل العمليات الإدارية. من هذه الوسائل استخدام برامج الدردشة الآلية (الشاتبوت) كمساعد للطلبة وموظفي الجامعة. تتيح برامج الدردشة الآلية الرد الفوري على استفسارات الطلاب وتوفر الدعم التقني والإداري على مدار الساعة مما يخفض التكاليف ويحسن تجربة الطلاب. اعتمادا على تكنولوجيات الذكاء الاصطناعي وتقنيات معالجة اللغات الطبيعية قمنا بانشاء روبوت دردشة آلي سميناه تطبيق الدردشة الجامعي UnivBot يسمح بالإجابة على الكثير من تساؤلات الطلبة ويخفف العبء عن أعوان الإدارة كي يتفرغوا لمهام أخرى. من مميزات هذا الروبوت أنه قابل للتطوير باستمرار عبر تحديث قاعدة المعارف المرتبطة به وتدريب نماذج اللغة وتقنيات التعلم العميق لتوفير إجابات دقيقة وسريعة في كل وقت.

## RESUME

À la lumière des développements technologiques rapides, les administrations universitaires cherchent à adopter de nouveaux moyens sophistiqués pour améliorer l'expérience des étudiants et faciliter les processus administratifs. L'un de ces moyens consiste à utiliser des chatbots comme assistants pour les étudiants et les employés de l'université. Les chatbots permettent de répondre immédiatement aux questions des étudiants et de fournir un soutien technique et administratif à temps plein, ce qui réduit les

coûts et améliore l'expérience des étudiants. En se basant sur les technologies d'intelligence artificielle et les techniques de traitement du langage naturel, nous avons développé notre propre chatbot appelé UnivBot qui permet de répondre à un grand nombre de questions des étudiants et de réduire le volume de travail important assuré par le personnel administratif afin qu'ils puissent se consacrer à d'autres tâches. Parmi les caractéristiques d'UnivBot, qu'il est capable d'être amélioré constamment et rapidement en mettant à jour sa base de connaissances, en utilisant des modèles de langage et des techniques d'apprentissage profond pour fournir des réponses en temps réel et avec une grande précision.