PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
IBN KHALDOUN UNIVERSITY - TIARET

Dissertation

Presented to :

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

For the Master's degree

Specialty: Software Engineering

With a view to creating a startup

Yusr

By:
Boulefred Yassmina
Boumaza Chaimaa

On the topic

A Context-Aware Chatbot for Customer Assistance Services

Soutenu publiquement le 11 / 06 / 2024 à Tiaret devant le jury composé de :

Mr LAAREDJ Zohra	MAA	IBN KHALDOUN UNIVERSITY	Chairman
Mr Abdelkader OUARED	MCA	IBN KHALDOUN UNIVERSITY	Supervisor
Mr Chikawi Ahmed	MCA	IBN KHALDOUN UNIVERSITY	Co-Supervisor
Mr BENAOUDA Habib	MAA	IBN KHALDOUN UNIVERSITY	Examiner
Mr Abdelkader BOUDALI		IBN KHALDOUN UNIVERSITY	Incubator Representative
Mr khlili Nadir	Employer in CNAC		Economic partner representative

2023-2024

Thanks

First of all, we thank Allah, the Almighty, who has given us the courage, strength, and will to do this work.

To our supervisor, Mr. Ouared Abdelkader, for all the time he has devoted to us , for his valuable advice and for all his help and support during the realization of this project.

To Mr.Chikaoui Ahmed , for all the add he give it to us for the realization of this project.

Also, to all the professors and staff of the computer science department to whom we owe our progress.

Finally, we also want to thank the members of the jury for accepting to evaluate our work.

Dedication

I dedicate this modest work to my beloved parents who supported and encouraged me until the end, to whom I owe all the love and respect.

To my dear sister and brothers.

To all my family.

To my dear friends and my partner Chaimaa.

To all the students of the Master GL promotion of the year 2023-2024.

And finally to those who are present in my heart.

Yassmina

Dedication

I dedicate this modest work to my dear parents who supported and encouraged me until the end, to whom I owe all the love and respect.

To my sisters .

To all my family.

To my dear friends and my partner Yassmina.

To all the students of the Master GL.

And finally to those who are present in my heart.

Chaimaa

Abstract

Today, in the face of increased competition between organizations, staying responsive and ready to address customer issues allows organizations to satisfy their customers, and participating in processes aimed at enhancing productivity. Universities and higher education institutions are organizations that provide educational services for students. These institutions witness a large number of students in higher education. Consequently, the vice-Dean in charge of studies and student affairs respond to various inquiries from the students. In addition, students' inquiries depend on their context, however the lack of these elements of the latter may exhibit discrepancies in terms of the student intent identification. To fully unlock the vice-Dean, we propose a Chatbot that provide more accurate and contextually relevant responses to students' queries. Our Chatbot is based rules with a context-aware that bridge the models of the user profile, context, and student inquiry to answer students' intents efficiently. The maturity of students' manifest and inquiries about their courses, and academic-related materials motivates us to go further, capitalize, and gather the most common questions (with their responses), we leveraged the ideas of a rule-based Chatbot, and we make it contextually aware to respond correctly. Tool support for the whole process is provided.

Keywords : Chatbot, Context-aware, Student Assistance, Conceptual modeling, Natural language processing, Semantic similarity.

Résumé

Aujourd'hui, face à une concurrence accrue entre les organisations, rester réactif et prêt à répondre aux problèmes des clients permet aux organisations de satisfaire leurs clients et de participer à des processus visant à améliorer la productivité. Les universités et les établissements d'enseignement supérieur sont des organisations qui fournissent des services éducatifs aux étudiants. Ces institutions accueillent un grand nombre d'étudiants dans l'enseignement supérieur. Par conséquent, le vice-doyen chargé des études et des affaires étudiantes répond à diverses demandes des étudiants. De plus, les demandes des étudiants dépendent de leur contexte, cependant, l'absence de ces éléments peut entraîner des divergences en termes d'identification de l'intention de l'étudiant. Pour décharger pleinement le vice-doyen, nous proposons un chatbot qui fournit des réponses plus précises et contextuellement pertinentes aux questions des étudiants. Notre chatbot est basé sur des règles avec une conscience contextuelle qui relie les modèles de profil utilisateur, de contexte et de demande de l'étudiant pour répondre efficacement aux intentions des étudiants. La maturité des étudiants et leurs demandes concernant leurs cours et les matériaux académiques nous motivent à aller plus loin, à capitaliser et à rassembler les questions les plus courantes (avec leurs réponses). Nous avons exploité les idées d'un chatbot basé sur des règles et nous l'avons rendu contextuellement conscient pour répondre correctement. Un support d'outils pour l'ensemble du processus est fourni.

Mots-clés : Chatbot, Sensible au contexte, Assistance aux étudiants, Modélisation conceptuelle, Traitement du langage naturel, Similarité sémantique.

ملخص

في ظل المنافسة المتزايدة بين المؤسسات اليوم، فإن البقاء مستجيبين وجاهزين لمعالجة قضايا العملاء يسمح للمؤسسات بتلبية احتياجات عملائها والمشاركة في العمليات التي تهدف إلى تعزيز الإنتاجية. تعتبر الجامعات والمؤسسات التعليمية العليا من المؤسسات التي تقدم خدمات تعليمية للطلاب. وتشهد هذه المؤسسات عددًا كبيرًا من الطلاب في التعليم العالي. وبالتالي، يستجيب نائب العميد المكلف بالدراسات وشؤون الطلاب للعديد من الاستفسارات من الطلاب. بالإضافة إلى ذلك، تعتمد استفسارات الطلاب على سياقهم، ومع ذلك فإن نقص هذه العناصر قد يظهر تناقضات من حيث تحديد نية الطالب. لتمكين نائب العميد بشكل كامل، نقترح روبوت دردشة يوفر ردودًا أكثر دقة وملاءمة للسياق على استفسارات الطلاب. يعتمد روبوت الدردشة الخاص بنا على قواعد مع الوعي بالسياق الذي يجسر نماذج الملف الشخصي للمستخدم، والسياق، واستفسار الطالب للرد على نوايا الطلاب بكفاءة. إن نضج الطلاب واستفساراتهم حول دوراتهم وموادهم الأكاديمية يدفعنا إلى المضي قدمًا والاستفادة وجمع الأسئلة الأكثر شيوعًا (مع إجاباتها)، واستفدنا من أفكار روبوت الدردشة القائم على القواعد، وجعلناه مدركًا للسياق للرد بشكل صحيح. يتم توفير دعم الأدوات لكامل العملية.

الكلمات المفتاحية: روبوت دردشة، مدرك للسياق، مساعدة الطلاب، النمذجة المفاهيمية، معالجة اللغة الطبيعية، التشابه الدلالي.

Table of Contents

Glossaire

— **AI :** Artificial Intelligence
— **NLP :** Natural Language Processing .
— **LLM :** Large language Model.
— **ALPAC :** Automatic Language Processing Advisory Committee.
— **NLU :** NAtural Language Understanding.
— **NLG :** Natural Language Generation.
— **LSTM :** Long Short-Term Memory .
— **RAG :** Retrieval Augmented Generation.
— **ASR :** Automatic Speech Recognition.
— **POS :** Part-of-Speech.
— **GPT :** Generative Pre-trained Transformer.
— **BERT :** Bidirectional Encoder Representations from Transformers.
— **SGD :** Stochastic Gradient Descent.
— **KG :**Knowledge Graph.
— **LU :** Language Understanding.
— **APIs :** Applications Programming Interfaces.
— **ANNs :** Artificiel Neural Networks.
— **LMS :** Learning Management System.
— **TF-IDF :** Term Frequency–Inverse Document Frequency.
— **JSON :** Java Script Object Notation.
— **NLTK :** Natural Language Toolkit.
— **ReLU :** Rectified Linear Unit.
— **UI :** User Interface.
— **UX :** User Experience
— **BMC :** Business Model Canvas.

Part I

General Introduction

Chapter

1 General Introduction

Contents

1.1 Introduction

1.1.1 Context and Motivation

Universities and other higher education institutions play a pivotal role in providing educational services quality to students [36, 35]. These institutions often accommodate a substantial number of students pursuing advanced education [44]. Consequently, the Vice-Dean, who is often in charge of overseeing academic matters and student affairs, has the responsibility of effectively addressing a diverse array of inquiries raised by students. Given the large number of students, a Vice-Dean cannot possibly serve all the inquiries personally. Therefore, relying on ChatBots that are able to process natural language may be an effective way of delivering information and instructions to students, or advice them in various areas like teaching/learning, campus information and locations, financial services, etc. [41].

1.1.2 Problem Statement

As a typical example, new students arriving on campus at the start of a new academic year may ask the same questions concerning teaching ("*Which courses are available this semester ?*" ; "*How do I register ?*" ; "*Who is my advisor ?*"), or financial ("*Am I eligible ?*" ; "*How do I apply ?*" ; "*Why haven't I received my financial aid ?*") matters. Aside from the number of inquiries, the answers to these questions usually depend on the individual circumstances and context of each student (e.g., personnal, demographics, academic information ; outcome data, but also emotion at the moment of the question), that are necessary to inform an adequate answer, to avoid discrepancies in accurately identifying the students' intents [5].

1.1.3 Our Contribution

This work tries to address the general research question of *How to bridge the models of the user profile, context and student inquiry to answer students' intents efficiently ?* In this project, we propose a framework centred around a context-aware Chatbot that is hybrid, meaning that it mixes rule-based principles and AI (typically, deep learning) models. This Chatbot fills a dashboard that provides a ranked list of appropriate responses to students inquiries, by rule-based principles to assist the Vice-Dean, The Chatbot provides a dashboard

It realigns the user profile, the specific context of a student and its inquiries with explicit specifications, to precisely define user intents. Our Chatbot analyse the students' context, and a ranked list of appropriate responses is recommended based on the students definition requirements. Thus, this Chatbot provides a dashboard that can be used by support staff and educational stockholders to always listen to these students and know what they want to join them. Notably, this Chatbot operates around the clock, 24 hours a day, 7 days a week, thereby providing continuous assistance to the Vice-Dean responsible for academic and student-related matters. Comprehensive tool support is dedicated to help universities and higher education institutions to cope with frequent questions of students and report the students' situations through automated support dashboard. Our solution be empowered by AI techniques (e.g., deep learning) that can be implemented as a designed pluggable component to realize a hybrid Chatbot, which constitutes a mixture between rule-based and AI models.

1.1.4 Thesis Organization

We start in chapter 2 by a few information about fundamental concepts on generative AI which are NLP and LLM. Chapter 3 Building Intelligent Conversational Agents we talk here about chatbots. Chapter 4 outlines our proposal A Context-Aware Chatbot for Student Assistance Services in Higher Education. Chapter 5 presents tool support, and on the Chapter 6 we talk about our Project Management then concludes and outlines future work.

Part II

Bibliographic Elements

Chapter

2 Fundamental Concepts on Generative AI

Contents

2.1 Introduction

Generative AI is a branch of artificial intelligence that involves creating models capable of producing new content that resembles human-generated output. One key area where generative AI has had a significant impact is natural language processing (NLP), the study and application of computational techniques to analyze, understand, and generate human language.

In the context of generative AI, large language models (LLMs) have emerged as transformative technologies. These models leverage vast amounts of data and sophisticated algorithms to understand and generate human-like text.

In this chapter we will introduces the foundational principles and methodologies underpinning these technologies.

2.2 Natural language processing

Natural languages are the languages which have naturally evolved and used by human beings for communication purposes. Natural Language Processing or NLP (also called Computational Linguistics) is the scientific study of languages from computational perspective. Natural Language Processing (NLP) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. Natural language generation systems convert information from computer databases into readable human language. Natural language understanding systems convert samples of human language into more formal representations such as parse trees or first order logic that are easier for computer programs to manipulate. [28]

The Figure 2.1 represent the typical workflow of NLP system . the system NLP work with a process

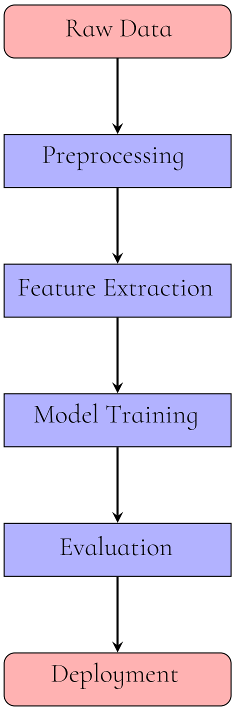


Figure 2.1 – Typical workflow of an NLP system.

2.2.1 History

In the early 20th century, Swiss linguistics professor Ferdinand de Saussure introduced the idea of "Language as a Science," emphasizing the systematic nature of language. Though Saussure died in 1913 without publishing his theories, his colleagues compiled his notes and students' notes into the foundational work Cours de Linguistique Générale in 1916, laying the groundwork for the structuralist approach in linguistics.

In the 1950s, Alan Turing proposed a test for determining a machine's ability to think like a human, while Noam Chomsky's Syntactic Structures revolutionized linguistics and grammar concepts, influencing AI and natural language processing (NLP). During this period, programming language LISP and early NLP programs like ELIZA were developed.

However, in 1966, the U.S. National Research Council's ALPAC committee halted funding for NLP and machine translation research, leading to a setback in AI and NLP research. This pause allowed for a re-evaluation of approaches, leading to a focus on statistical methods in the 1980s and 1990s. These methods, such as N-Grams and neural networks, greatly improved NLP's ability to handle large amounts of text data.

In 2011, Apple's Siri became one of the first successful NLP/AI assistants, using speech recognition to interpret voice commands and execute tasks. This marked a significant milestone in the evolution of NLP and AI technologies, paving the way for future advancements in these fields.[12]

Table 2.1 – Natural Language Processing Applications

Application	**Description**
Question Answering	**QA systems automatically answer questions in natural language.** [Image of Question Answering System]
Spam Detection	**Identifies and filters unwanted emails or messages.** [Image of Spam Email]
Sentiment Analysis	**Determines the emotional tone of text for opinion understanding.** [Image of Sentiment Analysis Graph]
Machine Translation	**Automatically translates text or speech from one language to another.** [Image of Machine Translation Interface]
Spelling Correction	**Identifies and corrects spelling errors in text.** [Image of Text with Spelling Correction]
Speech Recognition	**Converts spoken language into text.** [Image of Speech Recognition Interface]
Chatbot	**Conversational AI agents that simulate communication with humans.** [Image of Chatbot Conversation]
Information Extraction	**Automatically extracts structured information from unstructured text.** [Image of Information Extraction Process]

2.2.2 Levels of NLP

The most explanatory method for presenting what actually happens within a Natural Language Processing system is by means of the 'levels of language' approach. This is also referred to as the synchronic model of language and is distinguished from the earlier sequential model, which hypothesizes that the levels of human language processing follow one another in a strictly sequential manner. Psycholinguistic research suggests that language processing is much more dynamic, as the levels can interact in a variety of orders. Introspection reveals that we frequently use information we gain from what is typically thought of as a higher level of processing to assist in a lower level of analysis. For example, the pragmatic knowledge that the document you are reading is about biology will be used when a particular word that has several possible senses (or meanings) is encountered, and the word will be interpreted as having the biology sense.

Of necessity, the following description of levels will be presented sequentially. The key point here is that meaning is conveyed by each and every level of language and that since humans have been shown to use all levels of language to gain understanding, the more capable an NLP system is, the more levels of language it will utilize.

2.2.2.1 Phonology Level

This level deals with the interpretation of speech sounds within and across words. There are, in fact, three types of rules used in phonological analysis : 1) phonetic rules – for sounds within words ; 2) phonemic rules – for variations of pronunciation when words are spoken together, and ; 3) prosodic rules – for fluctuation in stress and intonation across a sentence. In an NLP system that accepts spoken input, the sound waves are analyzed and encoded into a digitized signal for interpretation by various rules or by comparison to the particular language model being utilized.[29]

Exemple 1. Phonology Level (Indirect)

Automatic Speech Recognition (ASR). An ASR system might analyze the phoneme sequence ("b", "æ", "t") in the spoken word "bat" to recognize it despite variations in pronunciation due to accent.

2.2.2.2 Morphological Level

This level deals with the componential nature of words, which are composed of morphemes – the smallest units of meaning. For example, the word preregistration can be morphologically analyzed into three separate morphemes : the prefix pre, the root registra, and the suffix tion. Since the meaning of each morpheme remains the same across words, humans can break down an unknown word into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffix –ed to a verb, conveys that the action of the verb took place in the past. This is a key piece of meaning, and in fact, is frequently only evidenced in a text by the use of the -ed morpheme.[29]

Exemple 2. Morphological Level

Identifying prefixes and suffixes. The system can recognize "un-" in "unhappy" as a negation prefix, indicating the opposite meaning of "happy".

2.2.2.3 Lexical Level

At this level, humans, as well as NLP systems, interpret the meaning of individual words. Several types of processing contribute to word-level understanding – the first of these being assignment of a single part-of-speech tag to each word. In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of-speech tag based on the context in which they occur.[29]

Exemple 3. Lexical Level

Part-of-speech (POS) tagging. The system assigns POS tags like noun, verb, adjective, etc., to each word in a sentence. (e.g., "The" - Determiner, "quick" - Adjective, "brown" - Adjective, "fox" - Noun, "jumps" - Verb)

2.2.2.4 Syntactic Level

This level focuses on analyzing the words in a sentence so as to uncover the grammatical structure of the sentence. This requires both a grammar and a parser. The output of this level of processing is a (possibly delinearized) representation of the sentence that reveals the structural dependency relationships between the words. There are various grammars that can be utilized, and which will, in turn, impact the choice of a parser. Not all NLP applications require a full parse of sentences, therefore the remaining challenges in parsing of prepositional phrase attachment and conjunction scoping no longer stymie those applications for which phrasal and clausal dependencies are sufficient. Syntax conveys meaning in most languages because order and dependency contribute to meaning.[29]

Exemple 4. Lexical Level

Dependency parsing. The system identifies the grammatical dependencies between words in a sentence, such as subject-verb or verb-object relationships. (e.g., "The quick brown fox jumps over the lazy dog" - "fox" is the subject, "jumps" is the verb, "dog" is the object)

2.2.2.5 Semantics Level

This is the level at which most people think meaning is determined, however, as we can see in the above defining of the levels, it is all the levels that contribute to meaning. Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. This level of processing can include the semantic disambiguation of words with multiple senses ; in an analogous way to how syntactic disambiguation of words that can function as multiple parts-of-speech is accomplished at the syntactic level. Semantic disambiguation permits one and only one sense of polysemous words to be selected and included in the semantic representation of the sentence.[29]

Exemple 5. Semantics Level

Sentiment analysis. The system determines the sentiment (positive, negative, neutral) expressed in a sentence, such as "This movie was fantastic !" (positive).

2.2.2.6 Discourse Level

While syntax and semantics work with sentence-length units, the discourse level of NLP works with units of text longer than a sentence. That is, it does not interpret multisentence texts as just concatenated sentences, each of which can be interpreted singly. Rather, discourse focuses on the properties of the text as a whole that convey meaning by making connections between component sentences. Several types of discourse processing can occur at this level, two of the most common being anaphora resolution and discourse/text structure recognition. Anaphora resolution is the replacing of words such as pronouns, which are semantically vacant, with the appropriate entity to which they refer. Discourse/text structure recognition determines the functions of sentences in the text, which, in turn, adds to the meaningful representation of the text.[29]

Exemple 6. Discourse Level

Coreference resolution. The system identifies pronouns and links them to the noun phrases they refer to in the text. (e.g., "Alice went to the store. She bought some milk.")

2.2.2.7 Pragmatic Level

This level is concerned with the purposeful use of language in situations and utilizes context over and above the contents of the text for understanding The goal is to explain how extra meaning is read into texts without actually being encoded in them. This requires much world knowledge, including the understanding of intentions, plans, and goals. Some NLP applications may utilize knowledge bases and inferencing modules.[29]

Exemple 7. Pragmatic Level *Speech act recognition. The system identifies the speaker's communicative intent in an utterance (e.g., question, request, command).*

Coreference resolution [a]. The system identifies pronouns and links them to the noun phrases they refer to in the text. (e.g., "Alice went to the store. She bought some milk.")

a. Coreference resolution (CR) is the task of finding all linguistic expressions (called mentions) in a given text that refer to the same real-world entity.

2.2.3 Preproccessing

Data Preprocessing is the most essential step for any Machine Learning model. How well the raw data has been cleaned and preprocessed plays a major role in the performance of the model. Likewise in the case of NLP, the very first step is Text Processing.

The Algorithm 2.1 shows a pseudo code of the Text Preprocessing in NLP process.

Algorithme 2.1 Text Preprocessing

```
Input : Raw text data
Output : Preprocessed text data
pour each document ▯ in text data faire
    Lowercase all text in ▯                              ◃Convert to lowercase
    Remove punctuation from ▯                            ◃Remove punctuation marks
    Tokenize ▯ into words                                ◃Split text into words
    Remove stop words from ▯                             ◃Remove common words
    (Optional) Stem or lemmatize words in ▯              ◃Reduce words to base form
    Return preprocessed text data =0
```

The various preprocessing steps that are involved are :

Lower Casing : It's quite evident from the name itself, that we are trying to convert our text data into lower case. But why is this step needed ?

When we have a text input, such as a paragraph we find words both in lower as well as upper case. However, the same words written in different cases are considered as different entities by the computer.

Tokenization : is the process of breaking up the paragraph into smaller units such as sentences or words. Each unit is then considered as an individual token. The fundamental principle of Tokenization is to try to understand the meaning of the text by analyzing the smaller units or tokens that constitute the paragraph.

Sentence Tokenize : we shall take a paragraph as input and tokenize it into its constituting sentences. The result is a list stored in a variable . It contains each sentence of the paragraph.

Word Tokenize : Similarly, we can also tokenize the paragraph into words. The result is a list called 'words', containing each word of the paragraph.

Punctuation Mark Removal : We must now remove the punctuation marks from our list of words.

Stop Word Removal : We must now remove the stop words which are a collection of words that occur frequently in any language but do not add much meaning to the sentences. These are common words that are part of the grammar of any language.

Stemming : is the process of reduction of a word into its root or stem word. The word affixes are removed leaving behind only the root form or lemma.

Lemmatization : Stemming does not always result in words that are part of the language vocabulary. It often results in words that have no meaning to the users. In order to overcome this drawback, we shall use the concept of Lemmatization.[43]

2.2.4 The components of NLP

NLP enables machines to read, understand, and interpret human language, an essential building block of many applications in various industries, such as customer service, healthcare, finance, and education.

The three components are key aspects of NLP :

1. **Speech recognition :** The translation of spoken language into text.

   Speech recognition, also known as Automatic Speech Recognition (ASR), converts spoken language into text. This technology enables computers to recognize and interpret human speech, which can be used in various applications, including virtual assistants, voice-enabled devices, and speech-to-text services.

2. **Natural language understanding :** A computer's ability to understand language.

   Natural language understanding (NLU) enables a computer to understand human language as it is spoken or written. NLU is a complex process involving multiple analysis layers, including syntactic, semantic, and pragmatic analysis.

   The syntactic analysis involves breaking down language into its grammatical components, such as sentences, clauses, and phrases. This stage involves identifying parts of speech, sentence structure, and other grammatical features that allow the computer to understand the language's syntax. Semantic analysis involves understanding the meaning of the language being used. This stage involves identifying the context, tone, and intent behind the language. It also involves identifying entities, such as people, places, and things, and their relationships to one another within the language.

   The pragmatic analysis involves understanding the social and cultural context of language use. This stage involves identifying social cues, such as sarcasm, irony, and humor, and understanding how these cues affect the meaning of the language.

3. **Natural language generation :** The generation of natural language by a computer.

Natural language generation (NLG) is the process of using computer algorithms to generate human-like language. NLG is a complex process that involves multiple layers of analysis and generation, including semantic analysis, sentence planning, and surface realization.

Semantic analysis involves understanding the meaning behind the information that needs to be conveyed. This stage involves identifying the relevant data, concepts, and relationships between them.

Sentence planning involves organizing the information into a coherent and meaningful structure. This stage involves determining the best way to present the information, such as selecting the appropriate sentence structure, tense, and voice.

Surface realization involves generating the actual text to be presented to the user. This stage involves applying the appropriate grammar and vocabulary to create a human-like sentence. [24]

2.2.5 Applications

Natural Language Processing plays an important role in various applications like Question Answering ,Spam Detection ,Sentiment Analysis ,Machine Translation ,Spelling correction ,Speech Recognition, Chatbot, Information extraction ...

2.2.6 Challenges

The field of Natural Language Processing (NLP) has made significant strides, yet numerous challenges persist. These challenges include :

Contextual Words and Phrases : Words and phrases can have multiple meanings based on context, which humans understand easily but pose difficulties for NLP models.

Synonyms and Complexity Levels : Synonyms and varying complexity levels of language (e.g., "large," "huge," "big") complicate the task of processing language, as different words may be used to convey the same idea.

Homonyms : Words that sound the same but have different meanings (homonyms) create problems in speech-to-text applications and question answering, since they rely on the text form to determine meaning.

Sarcasm and Irony : Detecting sarcasm and irony is particularly challenging for NLP models because such sentences can often be understood in the opposite way.

Ambiguity : Sentences that can be interpreted in more than one way pose a challenge for achieving high accuracy in understanding and processing.

Table 2.2 – Natural Language Processing Applications

Application	Description
Question Answering	**QA systems automatically answer questions in natural language.** [Image of Question Answering System]
Spam Detection	**Identifies and filters unwanted emails or messages.** [Image of Spam Email]
Sentiment Analysis	**Determines the emotional tone of text for opinion understanding.** [Image of Sentiment Analysis Graph]
Machine Translation	**Automatically translates text or speech from one language to another.** [Image of Machine Translation Interface]
Spelling Correction	**Identifies and corrects spelling errors in text.** [Image of Text with Spelling Correction]
Speech Recognition	**Converts spoken language into text.** [Image of Speech Recognition Interface]
Chatbot	**Conversational AI agents that simulate communication with humans.** [Image of Chatbot Conversation]
Information Extraction	**Automatically extracts structured information from unstructured text.** [Image of Information Extraction Process]

Informal Language : Informal phrases, idioms, and culture-specific language add to the difficulty of creating broad-use NLP models. These elements often vary greatly between different geographic areas and domains.

Domain-Specific Meanings : Words can have different meanings in various fields such as education, health, law, and defense, making it difficult to create models that perform well across different domains.

Misspelled or Misused Words : While autocorrect and grammar correction tools have improved, predicting a writer's intent, especially considering domain-specific, geographic, and informal language nuances, remains challenging.[26]

2.3 Large language model

Large language models are artificial intelligence (AI) systems that can understand and generate human language like text based on patterns and relationships learned from large amounts of data. Large language models use a machine learning technique called deep learning to analyze and process large amounts of data, such as books, articles, and web pages.

Exemple 8. Dialog Example with Gemini LLM *In this example, we asked Gemini for a suggestion for a future mobile application name : "Yusr". The power of Gemini is understanding the meaning in both English and Arabic.*

User Query : What does "Yusr" mean inArabic ?

Gemini Response : "Yusr" is an Arabic word that means "ease," ...

2.3.1 History

The idea for an LLM was first proposed in the 1960s with the creation of Eliza : the world's first chatbot, designed by MIT researcher Joseph Weizenbaum. Eliza marked the beginning of research in Natural Language Processing (NLP) and laid the foundation for future more sophisticated LLM.

Over the years, several significant innovations have propelled the field of LLMs forward. One such innovation was the introduction of Long Short-Term Memory (LSTM) networks in 1997, which allowed for the creation of deeper and more complex neural networks capable of handling more significant amounts of data.

Stanford's CoreNLP suite, introduced in 2010, was the next stage of growth allowing developers to perform sentiment analysis and named entity recognition.

In 2011, Google Brain was launched, providing researchers with access to powerful computing resources and data sets along with advanced features such as word embeddings, allowing NLP systems to better understand the context of words.

Table 2.3 – NLP vs. LLM : Comparaison

Aspect	NLP	LLM
Focus	Analysis	Generation
Goals	Understand	Create
Methods	Rules	Deep Learning
Interpretability	Clearer	Black Box
Applications	Specific Tasks	General Purpose
Data Needs	Varied	Massive

2.3.2 Architecture

Large Language Models (LLMs) are based on the transformer architecture, which has revolutionized natural language processing (NLP) since its introduction in the famous Attention is All You Need paper by Google researchers in 2017.

Transformer Architecture

The transformer architecture is considered the fundamental building block of LLMs. It is intended for neural networks to handle sequential data effectively.

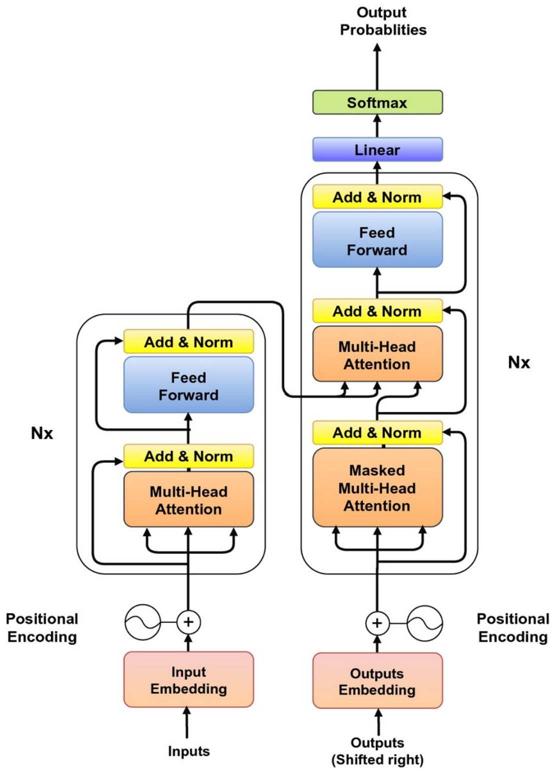


Figure 2.2 – The Transformer - model architecture.
[47]

Key components of the Transformer[15]

1. **Embedding Layer :**This converts input tokens (aka words) into vectors, which are numerical representations of the words along with semantic meaning and context of the text.
2. **Positional Encoding :**This step adds information about the position of each token in the sequence to its embedding. The purpose of this is to compensate for the lack of sequential processing in the Transformer.
3. **Encoder and Decoder :**The encoder processes input text and extracts context information, while the decoder generates coherent responses by predicting the next words in a sequence.
4. **Self-Attention Mechanism :**Self-attention allows the model to weigh the importance of different words in the input sequence. This enables it to understand context and relationships between words across the entire input sequence, which is essential for natural language where the meaning of a word can change based on its context within a sentence.

5. **Feedforward neural networks :**Both the encoder and decoder have feedforward neural networks which are responsible for applying additional transformations to the data processed by the self-attention mechanism. This enables them to capture even more nuanced features and context of natural language.
6. **Layer Normalization & Residual Connections :**Finally, these are techniques used within the model blocks to increase the training stability, avoid the vanishing gradient problem, and facilitate the training of deeper neural networks.

2.3.3 Popular Examples of Large Language Models

Pretrained language models play a pivotal role in natural language processing due to their ability to encapsulate broad language understanding and generation skills gleaned from diverse text sources. They offer a substantial advantage by minimizing the computational resources and data required for fine-tuning specific tasks. These are some of the most common pre-trained LLM models.[39]

Table 2.4 – Popular Large Language Models (LLMs)

Model	Developer	Year Launched
GPT-4	OpenAI	Confidential
Megatron-Turing NLG	NVIDIA	2022
Jurassic-1 Jumbo	AI21 Labs	2022
WuDao 2.0	BAAI	2021
WuDao 1.0	BAAI	2020
PaLM	Google AI	2022
Megatron-Turing NLG 530B	NVIDIA	2021
BLOOM	Hugging Face	2022
Gemini	Google AI	2022

GPT

Generative Pre-trained Transformer is an influential breakthrough in artificial intelligence, particularly in natural language processing (NLP). Developed by OpenAI, GPT leverages the Transformer architecture and extensive pre-training on vast internet text data to achieve a deep understanding of human language. This generative model excels at tasks like text generation, translation, question answering, and more, making it a versatile tool across various NLP domains. GPT's capacity to capture intricate language patterns and context, coupled with its iterative improvements, has profoundly impacted academia and industry, revolutionizing the landscape of language understanding and generation.

BERT

Bidirectional Encoder Representations from Transformers is a language model with a distinctive approach. Unlike previous models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by considering both left and right context in all layers. This pre-trained BERT model can be fine-tuned with minimal adjustments to create cutting-edge models for various tasks like question answering and language inference, eliminating the need for extensive task-specific modifications. BERT is both conceptually straightforward and remarkably effective.

T5

Text-to-Text Transfer Transformer is a groundbreaking large language model developed by Google Research, revolutionizing natural language processing (NLP). T5's innovation lies in framing all NLP tasks as text-to-text tasks, simplifying the NLP pipeline and unifying various tasks under a single framework. Built upon the Transformer architecture, T5 utilizes multi-head self-attention to capture intricate language relationships. Its extensive pre-training on vast text data, followed by fine-tuning on specific tasks, empowers T5 to excel in text classification, translation, summarization, question answering, and more. With consistently state-of-the-art results across NLP benchmarks, T5 has reshaped the field, offering researchers and developers a versatile tool for comprehensive language understanding and generation tasks.

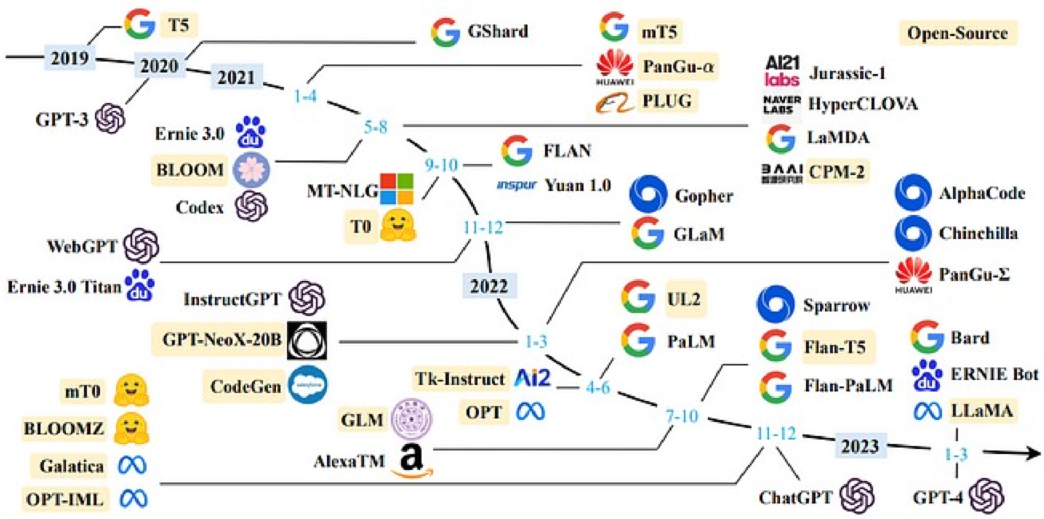


Figure 2.3 – A timeline of existing large language models [25]

2.3.4 Retrieval Augmented Generation

Large language models have achieved remarkable success, though they still face significant limitations, especially in domain-specific or knowledge-intensive tasks [23], notably producing “hallucinations” [51] when handling queries beyond their training data or requiring current information. To overcome challenges, Retrieval-Augmented Generation (RAG) enhances LLMs by retrieving relevant document chunks from external knowledge base through semantic similarity calculation. By referencing external knowledge, RAG effectively reduces the problem of generating factually incorrect content. Its integration into LLMs has resulted in widespread adoption, establishing RAG as a key technology in advancing chatbots and enhancing the suitability of LLMs for real-world applications.[16]

The Figure 2.4 illustrate the Retrieval-Augmented Generation (RAG) Model.

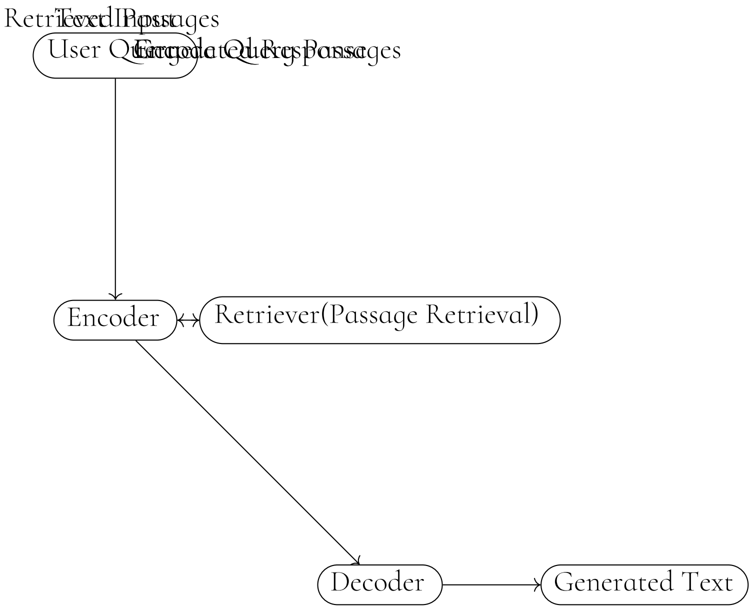


Figure 2.4 – Retrieval-Augmented Generation (RAG) Model

2.3.4.1 Types of RAG

The RAG research paradigm is continuously evolving, and we categorize it into three stages : Naive RAG, Advanced RAG, and Modular RAG.[16] the The Figure 2.5 shows the Comparison between the three paradigms of RAG (Gao et al. 2024).

Naive RAG

The Naive RAG research paradigm, which emerged soon after the adoption of ChatGPT, represents the earliest methodology in the RAG family. It follows a traditional "Retrieve-Read" framework comprising indexing, retrieval, and generation.

Indexing involves cleaning and extracting raw data from various formats (PDF, HTML, Word, Markdown) and converting it into plain text. This text is segmented into smaller chunks,

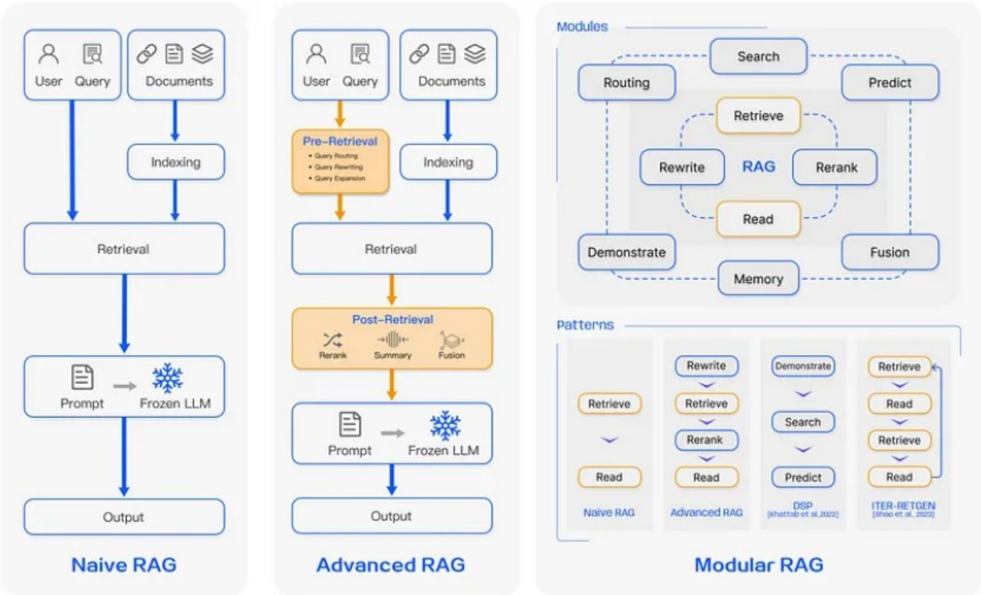


Figure 2.5 – Comparison between the three paradigms of RAG (Gao et al. 2024) [**<empty citation>**]

encoded into vector representations using an embedding model, and stored in a vector database for efficient similarity searches.

Retrieval occurs when a user query is received. The system encodes the query into a vector and computes similarity scores between this query vector and the stored text chunks, retrieving the top K chunks with the highest similarity.

Generation involves synthesizing the user query and retrieved documents into a prompt. A large language model then generates a response, either using its inherent knowledge or relying on the retrieved documents. In ongoing dialogues, the model can incorporate conversational history to support multi-turn interactions effectively.[16]

Advanced RAG

The techniques are divided into 4 high-level RAG components and innovations in each component :

- **Pre-Retrieval :** This phase involves preparing data and queries for efficient information retrieval. Techniques like search and ranking are discussed to steer the retriever towards retrieving relevant documents.
- **Retrieval :** In this phase, various approaches are employed to retrieve relevant information from documents. Examples : attention distillation, retrieval integration with reasoning, subgraph retrieval, and domain-specific summarization.

— **Post-Retrieval :** After retrieval, the focus shifts to re-ranking and filtering retrieved information to improve its quality and relevance. Methods like sequence-pair classification, iterative candidate selection, and knowledge distillation are discussed.
— **Generation :** This phase involves updating the generated content based on the retrieved information. Techniques like generating multiple retrieval queries, reward-driven context refinement, prepending retrieved documents, and self-consistency generation are discussed.

Modular RAG

The modular RAG architecture represents a significant advancement over the previous two RAG paradigms, offering greater adaptability and versatility. It incorporates a variety of strategies to enhance its components, such as adding a search module for similarity searches and fine-tuning the retriever. Innovations like restructured RAG modules and rearranged RAG pipelines have been introduced to address specific challenges. The trend towards a modular RAG approach is gaining momentum, supporting both sequential processing and integrated end-to-end training of its components. Although unique, Modular RAG builds on the foundational principles of Advanced and Naive RAG, demonstrating a progression and refinement within the RAG family.[16]

2.3.5 Training LLM

Training large language models involves several key steps that are fundamental to their successful development. The process typically begins with the collection and pre-processing of a large amount of text data from diverse sources , such as books, articles, websites, and other textual corpora. The curated dataset serves as the foundation for training the LLMs. After the removal of duplicates , noisy and poisonous data and ensuring privacy reduction , the training process involves unsupervised learning, where the model learns to predict the next word in a sequence given the preceding context assuming the language generation as a random process.

Currently, LLMs utilize Transformers which enable them to model long-range dependencies, understand text data enable them to generate new content in the style and characteristics of a genre or author. The training objective is to optimize the model's parameters to maximize the likelihood of generating the correct next word in a given context. This optimization is typically achieved through an algorithm called stochastic gradient descent (SGD) or its variants, combined with backpropagation , which computes gradients to update the model's parameters iteratively. [19]

2.3.6 Applications

Large language models (LLMs) have a wide array of applications across various fields due to their advanced natural language processing capabilities. Here are in Figure 2.6 some notable applications :[25]

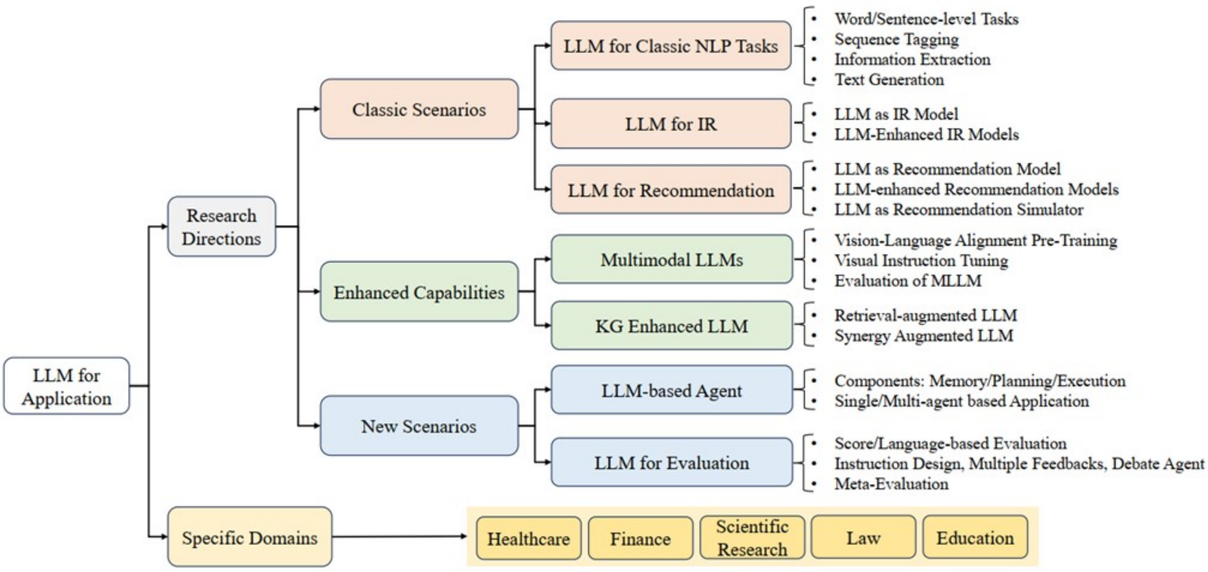


Figure 2.6 – Illustrates the various research directions and downstream domains where Large Language Models (LLMs) find applications

Classic NLP Tasks : LLMs handle traditional natural language processing tasks like word/sentence-level tasks, sequence tagging, information extraction, and text generation, offering robust solutions for diverse linguistic challenges.

Information Retrieval (IR) : LLMs are used both as models for retrieving information and to enhance existing IR models, improving the accuracy and relevance of search results.

Recommendation Systems : They serve in recommendation engines, both as primary models and as enhancements to existing systems, simulating recommendations, and aligning closely with user preferences.

Multimodal LLMs : These models integrate and process information from different modalities, such as text and images, enhancing interaction and understanding across various formats.

Knowledge Graph (KG) Enhanced LLMs : LLMs are augmented with knowledge from structured data sources like KGs, improving their ability to understand and generate contextually rich content.

LLM-based Agents : These agents, equipped with capabilities for memory, planning, and execution, can autonomously perform tasks, adapting to user requests and environmental feedback.

LLM for Evaluation : LLMs are used to evaluate and score other models or content, employing techniques like language-based evaluation and meta-evaluation.

Healthcare : Med-PaLM models achieved expert-level performance on the US Medical Licensing Examination, gaining approval from physicians for answering medical questions.

Education : ChatGPT helped students perform better in computer security courses by generating or refining answers.

Law : GPT-4 scored in the top 10% on a simulated bar exam, demonstrating its powerful legal interpretation and reasoning abilities.

Finance : BloombergGPT showed remarkable performance across various financial tasks, maintaining competitive performance in general-purpose tasks.

Scientific Research : LLMs assisted in various stages of the scientific research pipeline, including literature surveys, hypothesis generation, data analysis, and paper writing.

2.3.7 Challenges

The meteoric rise of Large Language Models (LLMs) in the field of machine learning has highlighted both their immense potential and the significant challenges they present. Let's delve into some of these challenges in more detail :

Data Complexity and Scale : The extensive datasets required to train LLMs are sourced from vast swathes of Internet text, making them nearly impossible to fully comprehend or scrutinize. This raises issues of data quality and inherent biases, leading to the potential spread of misinformation and harmful content.

Tokenization Sensitivity : LLMs depend on tokenization to break down text into manageable units. However, the choice and arrangement of tokens can dramatically affect the meaning and output of the model. This sensitivity can result in inconsistencies and vulnerabilities to adversarial attacks, where subtle changes in input yield significantly different outcomes.

Computational Resource Demands : Training LLMs is resource-intensive, requiring high-performance computing infrastructure and substantial energy. This not only limits accessibility but also contributes to significant environmental costs, raising concerns about the sustainability of such models.

Fine-Tuning Complexity : To tailor LLMs for specific tasks, fine-tuning is essential. This process involves creating task-specific datasets and extensive human annotation, making it time-consuming and expensive. The need for specialized datasets and manual effort adds another layer of complexity to deploying these models effectively.

Real-Time Responsiveness : Despite their advanced capabilities, LLMs often struggle with real-time responsiveness. Their inference speeds can be slow, which poses a problem for applications requiring immediate feedback, such as conversational agents and recommendation systems.

Contextual Constraints : LLMs are limited by the length of their context window, which restricts the amount of preceding text they can consider. This can make maintaining coherence in long documents or conversations challenging, as the model may forget or overlook crucial information from earlier in the text.

Bias and Undesirable Output : The biases present in the training data are reflected in the outputs of LLMs, leading to the potential generation of discriminatory, offensive, or harmful content. Addressing these biases is critical to ensure ethical and responsible AI deployment.

Knowledge Temporality : LLMs are trained on historical data and may not have access to the latest information. This temporal limitation can result in outdated or irrelevant responses, particularly when the conversation involves recent developments or events.

Evaluation Complexity : Current evaluation metrics for LLMs often fail to capture the full spectrum of model performance. These metrics can be manipulated, leading to misleading representations of a model's capabilities. Robust, nuanced, and comprehensive evaluation frameworks are necessary to accurately assess LLM performance.

Dynamic Evaluation Needs : Language is constantly evolving, and static evaluation benchmarks may not reflect a model's ability to adapt to changes in language and context. Therefore, evaluation frameworks need to be dynamic and continuously updated to stay relevant and accurate

2.4 Conclusion

We divide this chapter "Fundamental Concepts on Generative AI" into two fundamental secions wich are NLP where we talk about their levels, their components, some of their application and challenges. And LLM where we talk about their archeticture, the most popular exemples of LLM in now days, few informations about RAG, the training of LLM, some of their applications and challenges.

In the chapter below we will talk about intelligent conversational agents which means chatbots.

Chapter

3 Building Intelligent Conversational Agents

Contents

3.1 Introduction

With the use of AI developments like natural language processing, intelligent conversational agents, or chatbots, have revolutionized human-technology interaction. These agents, which have evolved from basic rule-based systems to sophisticated AI models, improve user experiences in industries like healthcare and customer service.

In these chapter we will introduces the foundational principles and methodologies underpinning the chatbots.

3.2 Chatbots

As per the Oxford English Dictionary, a chatbot is formally defined as follows :

Définition 1 (*Dialogue systems ?*)

> *A dialogue system is an artificial agent designed to interact with humans using (spoken or text-based) natural language.*

Définition 2 (*Chatbot*)

> *Chatbot (Noun) – "A computer program designed to simulate conversation with human users, especially over the Internet."*

A chatbot most commonly referred to as a smartbot, chatterbot, artificial conversational agent or interactive agent.[31]

Chatbots, chatterbots, Q&A agents or even Lingubots are software agents that offer access to a knowledge base through communication via an interface in natural language and an example of the application of AI to language. They are therefore computer programs that interact with users in natural language. Chatbots can be partly personified, which means that they can be displayed on the screen as humans, mythical creatures or animals. However, they can also appear as pure text input boxes.[46]

The Figure 3.1 represents the Basic architecture. This pipeline is commonly employed for chatbots, but it has a key limitation : it does not manage the overall dialogue beyond the immediate context. It is most suitable for brief interactions.

Natural Language Processing is what allows chatbots to understand your messages and respond appropriately. Natural Language Processing (NLP) helps provide context and meaning to text-based user inputs so that AI can come up with the best response.

As we can see in Figure 3.2, NLP is the main tool used for the chatbot to interpret the user's intent properly and accurately. Just like how AI is a broad and enormous field,natural language

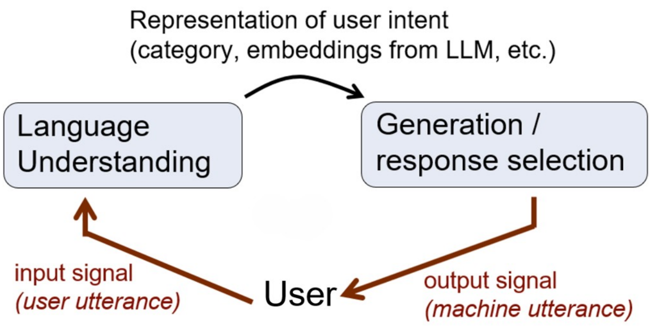


Figure 3.1 – Basic architecture

processing is also essentially an ocean of different algorithms used to convert text to important data for the chatbot to use.

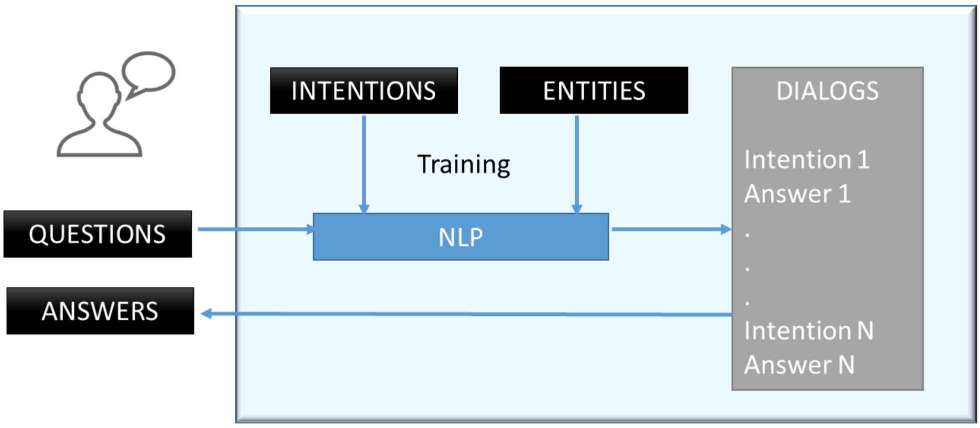


Figure 3.2 –Process of basic architecture

3.2.1 How it works ?

- **User Intercation :** Users can interact with the chatbot through various channels, such as a website chat window, messaging apps (e.g., Slack, Microsoft Teams), or voice assistants. Interaction can be initiated by sending messages or voice commands to the chatbot.
- **Natural Language Understanding (NLU) :** The chatbot employs Natural Language Processing (NLP) techniques to understand user queries and requests in natural language.
- **Meeting Assistance :** Meeting Assistance Customizable Responses : Users can customize the chatbot's responses to align with their communication style and preferences, ensuring a personalized experience.

AI Technologies Involved are : (i) Fuzzy Logic , (ii) Expert System, and (iii) Natural Language

Processing

3.2.2 Applications

Here is the main application of a chatbot :

— Mobile virtual assistants (Siri, Cortana, etc.)
— Tutoring systems
— Smart home environments
— In-car navigation & control
— Service robots

3.2.3 History of chatbots

The evolution of chatbots is a fascinating journey through time,   marked by significant milestones that reflect advancements in technology and shifts in human-computer interaction (Figure 3.3).

— 1950s : Turing Test throws down the gauntlet : Can machines chat like humans ?
— 1966 : ELIZA, the chatbot therapist, arrives to test the waters (with pattern-matching).
— 1970s : Chatbots like Parry get specific, delving into focused topics.
— 1988 : Jabberwacky the chatbot crashes the video game scene !
— 1990s : A.L.I.C.E. chats it up with natural language processing.
— 2000s : SmarterChild becomes your virtual buddy on IM.
— 2010 : Siri redefines personal assistants on Apple devices.
— 2016 : Facebook Messenger Bots open a new chapter in social media chat.
—2022: ChatGPT, a large language model chatbot developed by OpenAI, is introduced.

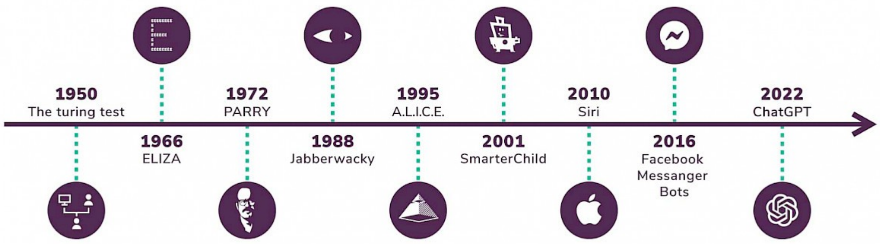


Figure 3.3 – History of chatbots

3.2.4 Types of chatbots

Chatbots can be classified into different types based on the knowledge they have or which they access, their level of interaction, their method of response generation,their service provided or the goal they want to achieve.

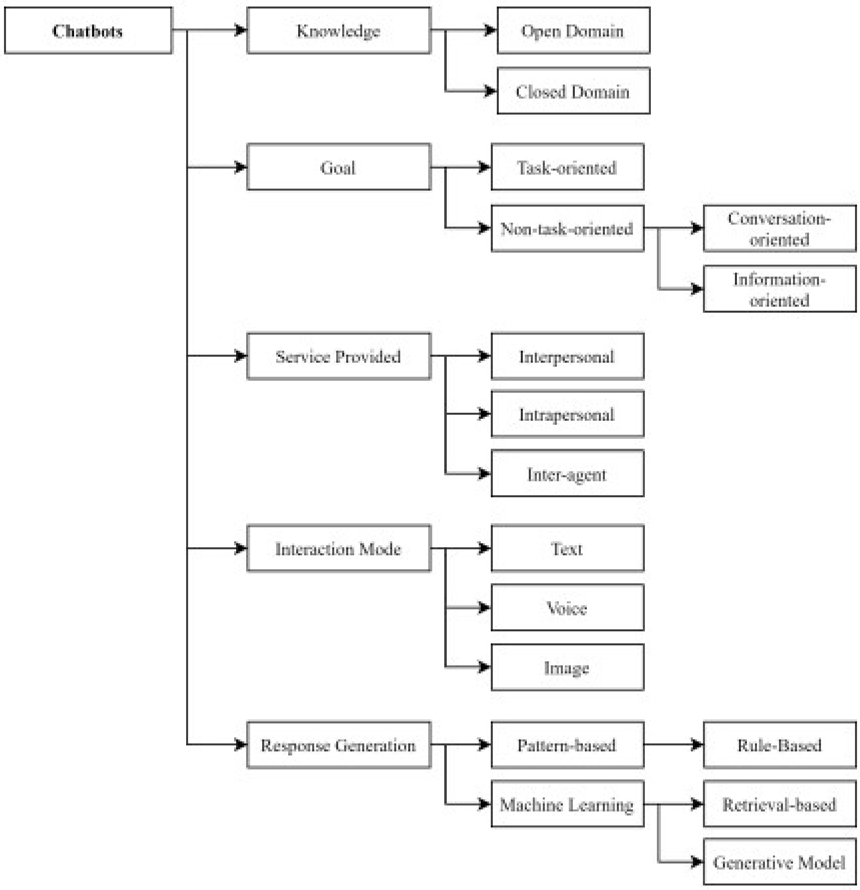


Figure 3.4 – Classification of Chatbots[11]

3.2.4.1 Knowledge Domain

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

3.2.4.2 Goal

Chatbots fall into two main categories : task-oriented and non-task-oriented.

— Task-oriented chatbots excel at completing specific tasks but lack general knowledge. They require custom training data and have a predetermined conversation flow.
— Non-task-oriented chatbots aim for natural conversation. They can be generative (creating new responses) or retrieval-based (selecting pre-made ones). These chatbots require a massive amount of training data and struggle with specific tasks.this chatbot have two sub-types : Information-oriented : similar to FAQs, retrieves pre-stored information. Conversation-oriented : aims for continuous, engaging dialogue.

3.2.4.3 Service Provided

This approach categorizes chatbots based on the service they offer and the user interaction they facilitate. It considers three factors :

— Sentimental Proximity : How close the emotional connection between the user and the chatbot is.
— Level of Intimacy : The degree of personal interaction involved in the conversation. Task Performance : The specific task the chatbot is designed to accomplish. Here's a breakdown of the three main types :
— Interpersonal Chatbots :

Focus : Communication and task completion. Examples : Restaurant booking, flight booking, FAQ bots.

3.2.4.4 Interaction Method

Chatbots come in two flavors : text-based and voice-based.
— Text-based chatbots , handle written messages, ideal for desktops, mobiles, and messaging apps. They're well-suited for complex tasks involving logic or confirmations, where users can control the flow of conversation.
— Voice-based chatbots , like Siri or Alexa, use spoken voice for interaction. Perfect for multitasking, they allow hands-free control for activities like playing music or managing smart home devices[11].

3.2.4.5 Response Generation

This table dissects chatbot response generation, contrasting pre-programmed rules with data-driven approaches. Machine learning shines, offering retrieval for efficient responses or

Chatbot Response Generation	Classification	Description
Rule-based	Predefined rules and keywords	Simple to set up, but inflexible
Machine Learning (ML)	Analyze large amounts of data to learn	More flexible and adaptable
Retrieval-based	Search for most relevant response in database	Efficient, but response quality depends on data
Generative	Generate new responses on the fly	More natural conversations, but requires more data and power

Table 3.1 – Classification of Chatbot Response Generation

generative power for natural conversations, but demands more data and resources

Rule-based approach, rulebased chatbots function like a decision tree, relying on predefined rules and keywords. They excel in simple scenarios with limited answers (e.g., QA bots) but struggle with complex conversations or learning. Their repetitive, pre-programmed responses lack human-like engagement, making them unsuitable for entertainment. While effective for limited tasks with foreseeable outcomes, they can't adapt or learn from interactions.

Retrieval-based approach , the chatbots who base on this appeoach offer more flexibility compared to rule-based models. They leverage APIs to access and analyze information, providing fluent and grammatically correct answers within a specific domain. This is achieved through pattern-matching or Machine Learning to select the best pre-defined response. However, they lack context awareness, cannot reference past interactions, and struggle with user input outside their domain. While predictable responses can be helpful for tasks like product information, they hinder casual conversation or handling grammatical errors. Additionally, they require a rich dataset and often use Artificial Neural Networks for scoring potential answers.

Generative approach , the most sophisticated approach, create new responses based on past interactions, fostering a human-like experience. However, they require massive datasets for training, leading to unpredictable outputs and potential biases. This makes them ideal for casual conversation but less suitable for task-oriented applications [11].

3.2.5 General Architecture

When a request is received by the system, the Language Understanding (LU) component is in charge of interpreting the user's words and their meaning. Then, the Dialog Management component either formulates a response or if necessary asks users for further clarifications. The Dialog Management component also interacts with web services and knowledge sources to collect the required information to include in the response. The actual component that then constructs the response is called Response Generation (RG) .

This representation is suitable for a system that recognises text input. If it were to also recognise voice input, it would lack the step of speech recognition before the Language Understanding step, and the step of text-to-speech synthesis after the Response Generation step for converting audio into words and words into audio back again.

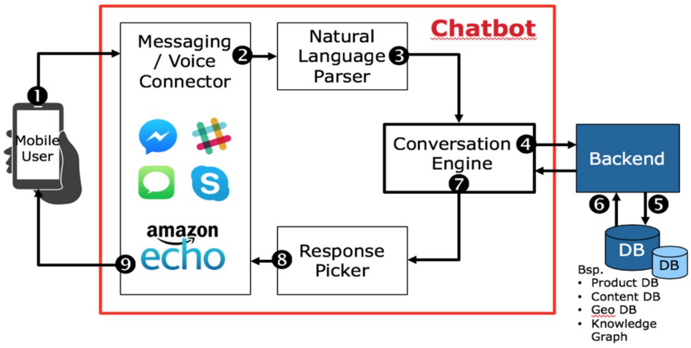


Figure 3.5 – General Architecture of chatbot [3]

As we see in figure 3.2 , The process commences when a user submits a request to the chatbot through a messenger app such as Facebook, Slack, WhatsApp, WeChat, or Skype, or through an app that accepts text or speech input like Amazon Echo.

Upon receiving the user's request, the Language Understanding Component analyzes it to determine the user's intention and the relevant information associated with it (intent : "translate," entities : [word : "environment"]).

Once the chatbot has reached its best interpretation, it must decide how to proceed. It can directly act upon the new information, retain the understanding and wait for further developments, request additional context, or seek clarification. Once the request is understood, the chatbot carries out the requested actions or retrieves the desired data from its data sources, which could be a database known as the chatbot's Knowledge Base, or external resources accessed through an API call. After retrieving the information, the Response Generation Component utilizes Natural Language Generation (NLG) to generate a human-like response in natural language, based on the user's intention and the context information obtained from the analysis of the user's message. The appropriate responses are generated using one of the three models mentioned in the paper : rule-based, retrieval-based, or generative model. A Dialogue Management Component maintains and updates the conversation context, which includes the current intention, identified entities, or any missing entities required to fulfill the user's requests. Additionally, it prompts for missing information, processes user clarifications, and asks follow-up questions. For instance, the chatbot may respond : "Would you also like to provide an example sentence using the word 'environment' ?"[1].

3.2.6 Challenges

Chatbots, while increasingly prevalent and sophisticated, face several challenges.Some of these challenges are [11] :

- **Personal perception about performance :** Users' subjective opinions about how well a system performs can vary greatly. These perceptions can be influenced by factors such as previous experiences with similar technologies, individual expectations, and the specific tasks they need to accomplish. A system might objectively perform well, but if users perceive it as slow or cumbersome, their overall satisfaction will be negatively impacted.
- **Keeping the system interactive :** An interactive system is one that responds quickly and effectively to user inputs, maintaining engagement and facilitating smooth interactions. Ensuring interactivity is crucial for keeping users involved and preventing frustration. This involves minimizing response times, providing immediate feedback, and creating a seamless user experience.
- **Data Privacy :** Protecting user data is a critical concern in any system. Users need to trust that their personal and sensitive information is secure from unauthorized access and breaches. Ensuring data privacy involves implementing robust security measures, complying with regulations, and being transparent about how data is collected, stored, and used.
- **Difficulty of the training :** Training users to effectively use a new system can be challenging. This difficulty can arise from the complexity of the system, the need for specialized knowledge, or insufficient training resources. If the training process is too demanding or inadequate, users may become frustrated and the system's adoption and utilization may suffer.
- **Applying NLU techniques considering the importance of semantics :** Natural Language Understanding (NLU) techniques must accurately interpret the meaning and context of user inputs. Semantics play a crucial role in understanding user intent and providing appropriate responses. Ensuring that NLU systems can effectively handle the nuances of language is vital for creating intuitive and effective interactions.

3.3 Related work and our positioning work

The chatbots have demonstrated their efficacy across various domains, including the banking sector, universities, and telecommunications, as evidenced by multiple studies (see surveys by [27] and [2]). This section reviews works focused on developing context-aware chatbots for student assistance services in higher education.

The development of chatbots has become increasingly important as they enable companies to offer 24/7 customer service while reducing costs, permeating diverse areas such as education (e.g., [45, 49, 50]).

Moreover, several rule-based chatbots have been proposed ([9, 42, 42, 37, 48]). For example, [42] explores using chatbots to support student goal setting and social presence in fully online activities, enhancing learner engagement and perceptions. [9] discusses components of a smart chatbot academic model for university websites. Additionally, [40] highlights the implementation of Telegram as a pedagogical tool to enhance student motivation and interaction.

In this section, we focus on chatbots designed with context-awareness in mind. Similar to our work, early initiatives have proposed context-aware chatbots for various purposes, such as tourism, to recommend appropriate services to end users (e.g., [9]). [5] introduces hierarchical recurrent attention networks for context-aware educational chatbots, while [20] proposes a FAQ chatbot for inclusive learning in massive open online courses. Aligned with our approach, [18] presents the Context-Aware Self-Attentive Natural Language Understanding (CASA-NLU) model, leveraging various signals within a flexible context window, including prior intents, slots, dialog acts, and utterances alongside the current user input.

Similar efforts have been made by the recommendation systems community (e.g., [9, 32, 22, 4]) to develop intelligent chatbots that provide content-based recommendations tailored to user requirements. For instance, [9] offers recommendations for content and services tailored to tourist profiles and their immediate context, assisting visitors during their exploration of cultural sites.

Although significant work has been done on chatbots in education (e.g., [33, 8, 7]), previous studies have not sufficiently focused on context-aware chatbots for student assistance by integrating user profiles, context, and intent. Establishing such a connection is crucial for designing chatbot models that contextually address user queries concerning intent recommendations within the assistance services domain.

3.4 Conclusion

This chapter covers the history and current relevance of chatbots, or intelligent conversational agents, which mimic human speech using artificial intelligence. It explains their types, general architecture, and evolutionary history in historical context it also addresses challenges of chatbots.

In the chapter below we will talk about our solutions which is "A Context-Awre Chatbot for Student Assistance Services in Higher Education".

Part III

Contribution

Chapter

4

A Context-Aware Chatbot for Student Assistance Services in Higher Education

Contents

4.1 Introduction

In today's highly competitive environment, organizations must adopt a proactive and steadfast approach to addressing client concerns. This strategy not only ensures client satisfaction but also boosts overall productivity. Similarly, the business vocabulary is increasingly permeating higher education, giving rise to the concept of the "economic university" where students are seen as customers. Consequently, universities must be prepared to provide support and assistance to effectively satisfy, retain, and motivate their students.

This chapter delves into the development of context-aware chatbot designed to assist students with various academic and administrative tasks.

4.2 Motivating Example

In this section, we present our motivating example.

4.2.1 Scenarios : Higher education massification

Currently, we are witnessing a massification of the number of students in institutions and universities. As a result, administrators and technicians tasked with addressing these requests see their workload increase due to the various inquiries from students. However, to fully unlock these users, it becomes necessary to employ a method that remains active to listen to students (24 hours and 7 days.).

The Figure 4.1 represent the massification in higher education and the load for the administration.

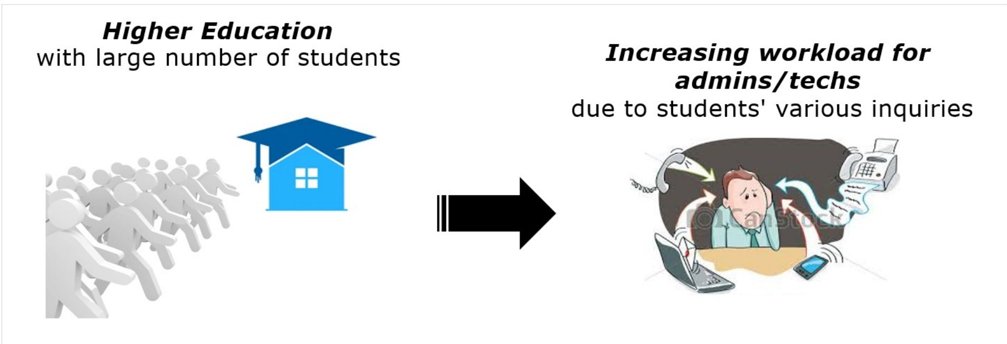


Figure 4.1 – Scenarios : Higher education massification

The ongoing trend of burgeoning student populations within educational institutions and universities presents a multifaceted challenge for administrators and technicians tasked with managing the influx of inquiries and requests from these individuals. As the numbers swell, so too does the workload of those responsible for addressing the diverse array of needs and concerns that students bring forth.

In this landscape of expanding educational enrollment, it becomes increasingly apparent that a proactive and accessible method for engaging with students is imperative. The traditional modes of communication, such as office hours or email correspondence, are no longer sufficient to accommodate the round-the-clock nature of student life and the varied schedules of individuals within the academic community.

To fully unlock the potential of these users and ensure their needs are met in a timely and efficient manner, it becomes necessary to embrace a method that remains active and accessible 24 hours a day, 7 days a week. This entails implementing robust and dynamic systems that facilitate continuous communication and support, regardless of the time or day of the week.

4.2.2 Domain Analysis

This part represents an important milestone in the completion of the project. We can divide this stage into four sections :i) Collection , ii) analysis, iii) classification and iii) integration. First, we gathered a collection of different files by studying the environment and the subject. We sought assistance from both the university staff, including professors and administrators, as well as some students and various documents. These files were then processed and classified according to different fields such as registrations and examinations. Finally, they were integrated into a single consolidated database.As we can see in Figure 2.2

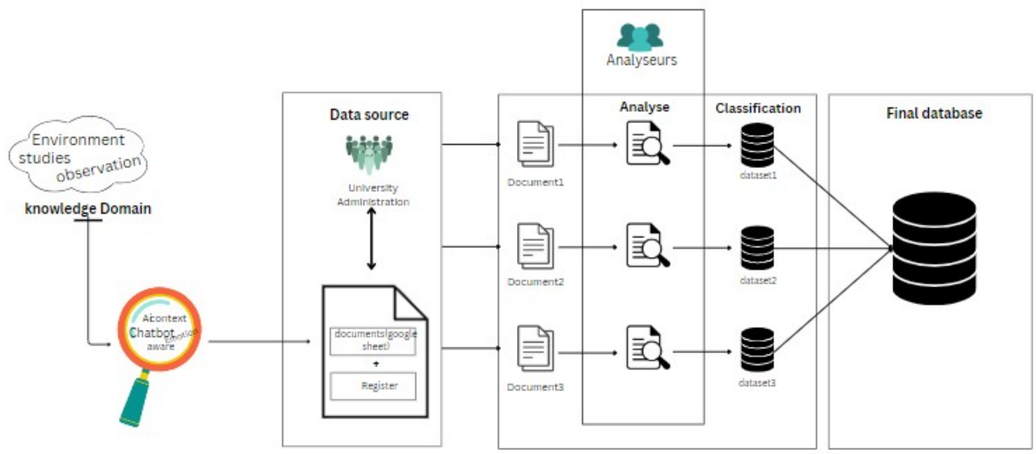


Figure 4.2 – Domain analysis

4.2.3 Motivation and Research Question

As we see in Figure 4.3 the students' intents depend on their Profile,Context and Inquiry type . In our case the user profile contains personal information (e.g., name, age, education history, interests, contact information) and their interactions across various social media platforms such as twiter, as well as other platforms like learning management systems (LMS) such as Moodle.

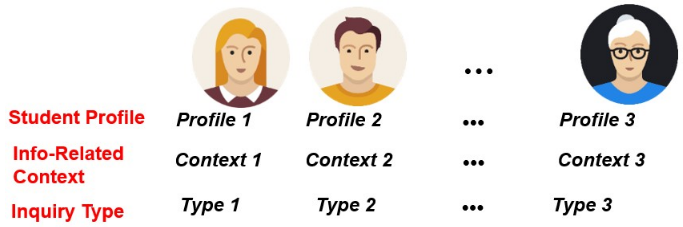


Figure 4.3 – Motivation and Research Question

"context" represents all the circumstances surrounding the student's situation.

An "inquiry“represents the type of requirement, which can be advice, a manifest, assistance, etc.

To explain more this challlenges, as we can see in Figure 4.4.Consider Sarah an international, first-year bachelor student living on campus, who likes extracurricular activities and has consequently joined the coding science club. As a context, Sarah is classified with a category preference (extracurricular activities, with coding as a subcategory), and a demographic (international) and academic (first year) information. Samia is an international second-year bachelor degree with a state scholarship, who asks financial services "How is it possible to open a bank account ?". Samia's inquiry may be related to her scholarship situation, meaning that understanding that "a bank account" may represent for her the ability to receive the scholarship payments from her country on a national account. The system may answer with information on how to open accounts remotely, and what paperwork is required from the bank to do so. This is called context awareness : it enables the possibility to understand students' followup questions, and to relate those with previously served information to provide users relevant and valuable answers.

For instance, in the case of the student Sarah, a spatial search of students' context would retrieve all of the services and inquiries associated with the categories of extracurricular activities and preferences related to their sub-categories. Many inquiries may not be related to coding or extracurricular activities, leading to low query result precision

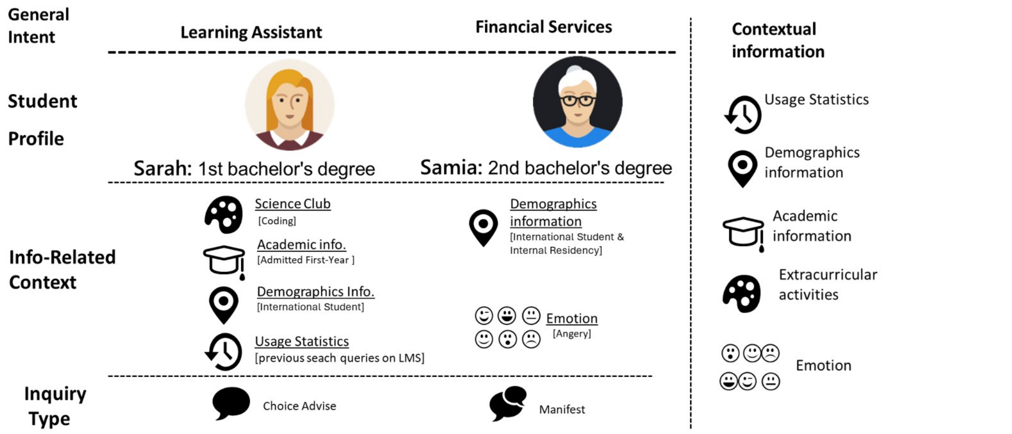


Figure 4.4 – Motivating Example

From these examples, one can see that the Context Parameters depend on set of parameters that can be derived from different dimensions (like Personal Information, Demographic Information, Academic Information, Outcomes, Extracurricular Activities, and Emotions). See Figure 4.5.

On the other hand, the students ' intents can be organized into Intent classes and their categories (Figure 4.5). we argue that all students intents fall into one of these categories and ease the way to identify the context correctly.

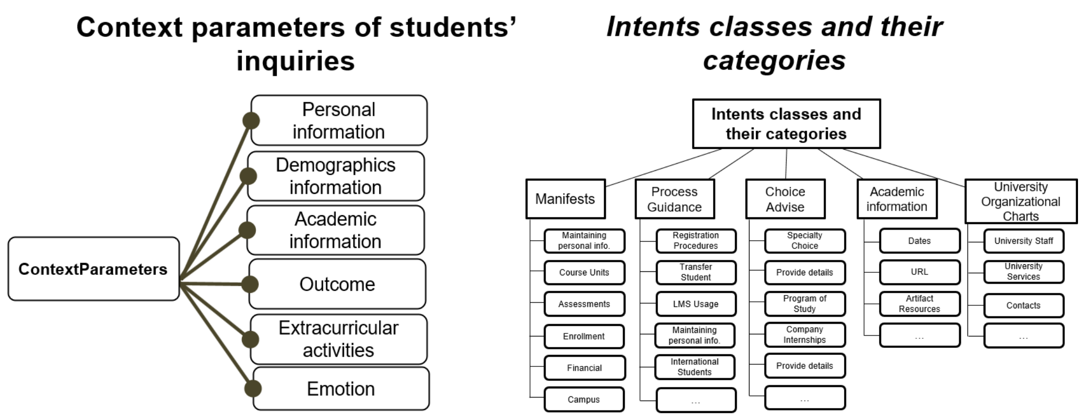


Figure 4.5 – Context parameters of students' inquiries, and Intents classes and their categories

In this case, we have identified five classes of intent : Manifests, Process Guidance, Choice Advice, Academic Information, and University Organizational Charts.

From the above , we can extract the following research question :

— *How to bridge the user profile,context and student inquiry to answer students' intents efficiently ?*

4.2.4 Our Vision

To resolve the problem, we have proposed to explicitly define the relationship between context, profile, and contextual information. Our vision is to make clear the connection between all this information. As you can see in this figure, the core element is the intent, which is associated with context. The context provides a set of entities, and the intent represents students' inquiries and has a set of patterns of "Predefined Rules" with associated responses. We have proposed in the next section a metamodel that explicates all these components for each instance of intent. This Metamodel presents our conceptual model of our Chatbot's main components used in the control flow logic.

The Figure 4.6 shows the vision of our soulution that connect all the component of our chatbot explicitly . The left side of Figure 4.6 present the instence of the privieus motivating by using these componenent.

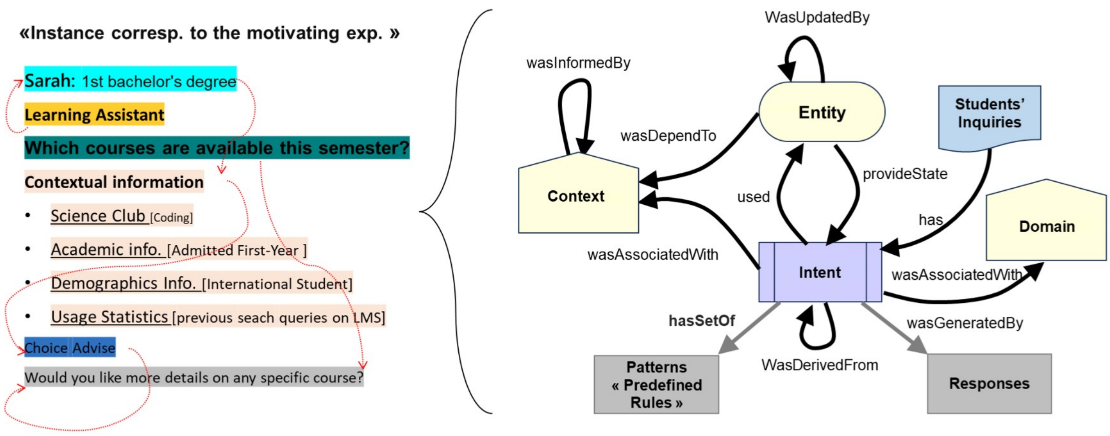


Figure 4.6 – Our Vision

4.3 Our Proposal

We propose Yusr, a chatbot combining rule-based and AI approaches, designed to provide quick and efficient answers to students' inquiries regarding their university life. Yusr reduces the manual effort of repeatedly answering the same questions by integrating contextual information and previous chat discussions to provide more focused responses. This frees up time and resources. Support staff can also benefit from a dashboard that provides additional information about students' inquiries.

4.3.1 Hybrid Chatbot Services

Our solution is a context-aware chatbot to handle student queries, which is an advanced artificial intelligence system designed to help students by understanding their questions and answering them in a smarter and more contextually appropriate way.

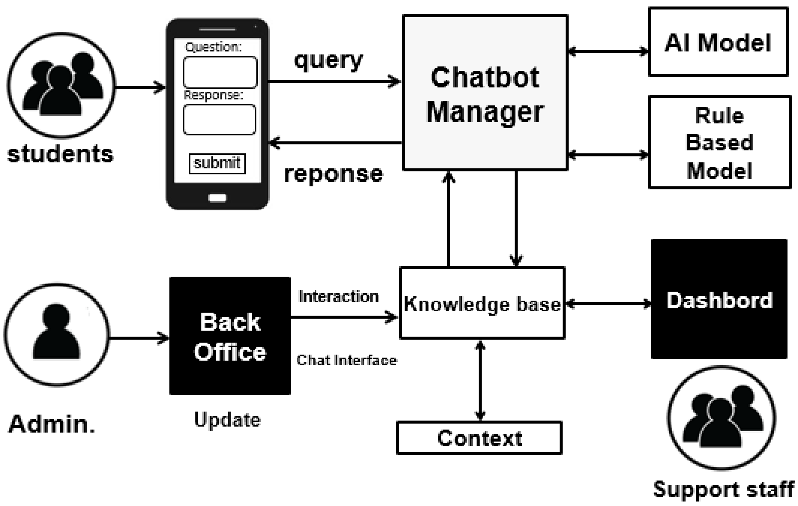


Figure 4.7 – Overview of our proposal

As we see on the overview in Figure 4.7, the process begins when the user (the student in this case) sends a request to the chatbot through the Yusr messaging application, and the chatbot manager, who manages and processes the group of requests, receives the sent request and then sends it either to Rule Based Model or AI Model which directs the generated answer to the knowledge base that contains all the information about the current situation controlled by the back office management via, which also depends on the context. Finally, the answer is returned to the chat manager. A bot that processes it and outputs it as an accurate response to the sent request. The frequency and trends of student inquiries are also monitored by the support staff service through the support staff service dashboard. We can say that the Hybrid Chatbot program contains the following basic services :

- Chatbot Manager : to manage and manipulate the set of messages.
- Response Generator : that is based on Rule Based Model and AI Model.
- Knowledge Base : that contain all information about current situation.
- Back Office Admin : this services is dedicated to users to introduce the current information about the context.

— Support Staff Dashboard : this services is dedicated for support staff to monitor the frequency and trends of students' inquiries.

4.3.2 High-Level Architecture

Our system is composed of three main components :

— **NLP Preprocessing Unit :** in the first step, the query expressed by students in NL (Natural Language) be passed through the pre-processing phase (i.e. tokenization, Stop-words removal, POS tagging (Part-Of-Speech tagging) and lemmatization.

— **Interpretation Unit :** In order to define the students' intent, our system is based on three catalogs to determine respectively Entities using Entity Classifier, Target Domain using Domain Classifier, and Query Context using Context Classifier. A transformation to an SQL query should be launched in order to select the elements that fits the student's requirement from these catalogs. After identifying the user intent, our system used a matcher based on the similarity to return the appropriate responses from the Response Dictionary (see Figure 4.8).

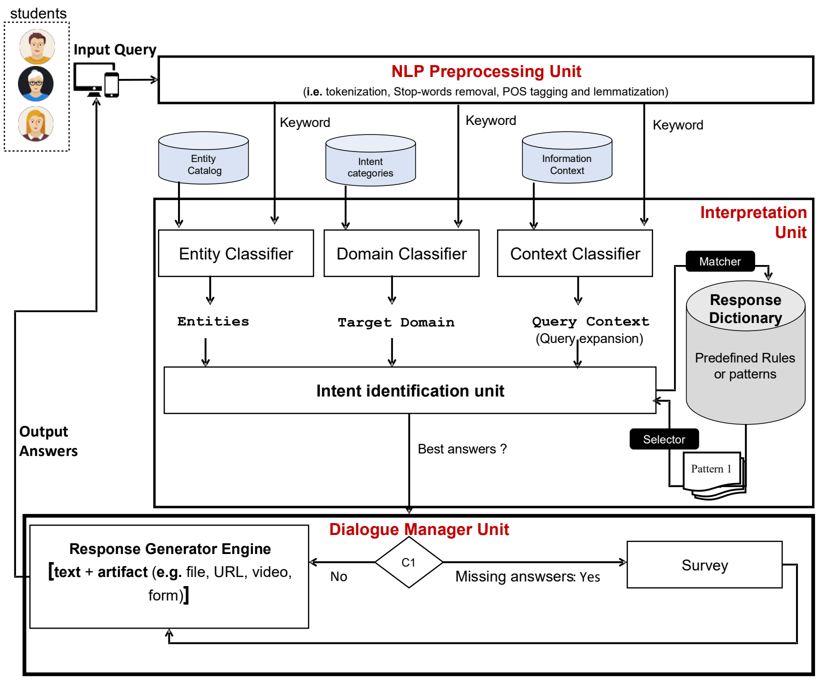


Figure 4.8 – High-Level Architecture.

4.3.3 Conceptual Organization

Figure Figure 4.9 presents our conceptual model of our Chatbot's main components used in the control flow logic.

A `ChatbotAgent` is customized by a set of parameters using a `name`/`value` configuration, and contains a sequence of query/response organized as a `ResponseDictionary` : each entry references an intent corresponding to an `InputQuery` , and a set of responses stored with its corresponding resources (text, image, video). An `Intent` generalizes a `Phrase` , an `Event` , and an `Action` (i.e. confirming inputs) which supports basic interactions (e.g., by proactively presenting the available options). An `Intent` is matched to an `Entity` , thanks to a `MatchingIntent` . Also note that an `Entity` is expanded using the current `Context` to identify with precision the target intents. A `Context` captures several pieces of information : an `AcademicInformation` and `PersonalAttribute` s, which constitutes the student's profile ; the student's educational journey as an `Outcome` ; the `DemographicInformation` ; `ExtracurricularActivities` ; and the `Emotion` reflecting the emotional state.

To make a recommendation, a `Survey` represents a list of questions that can be retrieved either by asking a `Question` to a student, or automatically *via* an `Analysis` of the Chatbot model, which can then be extended according to the system constraints and user requirements.

In order to improve performance for responses, inquiries are classified according to separate categories (which include their own subcategories) that help prune the search space : Manifest, Process Guidance, Choise Advice, Academic Information, and University Organizational Charts.

Finally, we use a set of similarity-based matching algorithms to relate the student's context (`MatchingContext`), intent (`MatchingIntent`) and inquiry (`MatchingInputQuery`) to improve the precision of keyword extraction and locate appropriate responses (cf. **? ?** for details).

4.3.4 Conversational Workflow

Here's in Figure 4.10 an example to illustrate the Conversational Workflow :

— For instance, a student asks the following question, "Can you tell me what the deadline is for submitting the final project ?"
— First, the Chat Manager (Message manipulation : just received a message from) receives the message and forwards it to the rules-based chatbot or the AI model.
— Our chatbot use a response repository organized according to our metamodel to retrieve one or more responses.
— The response is then communicated back to the Chat Manager, which incorporates this response into the context using a Knowledge Base that is periodically updated using School Services.
— Now, let's zoom in on the components of the rules-based chatbot.

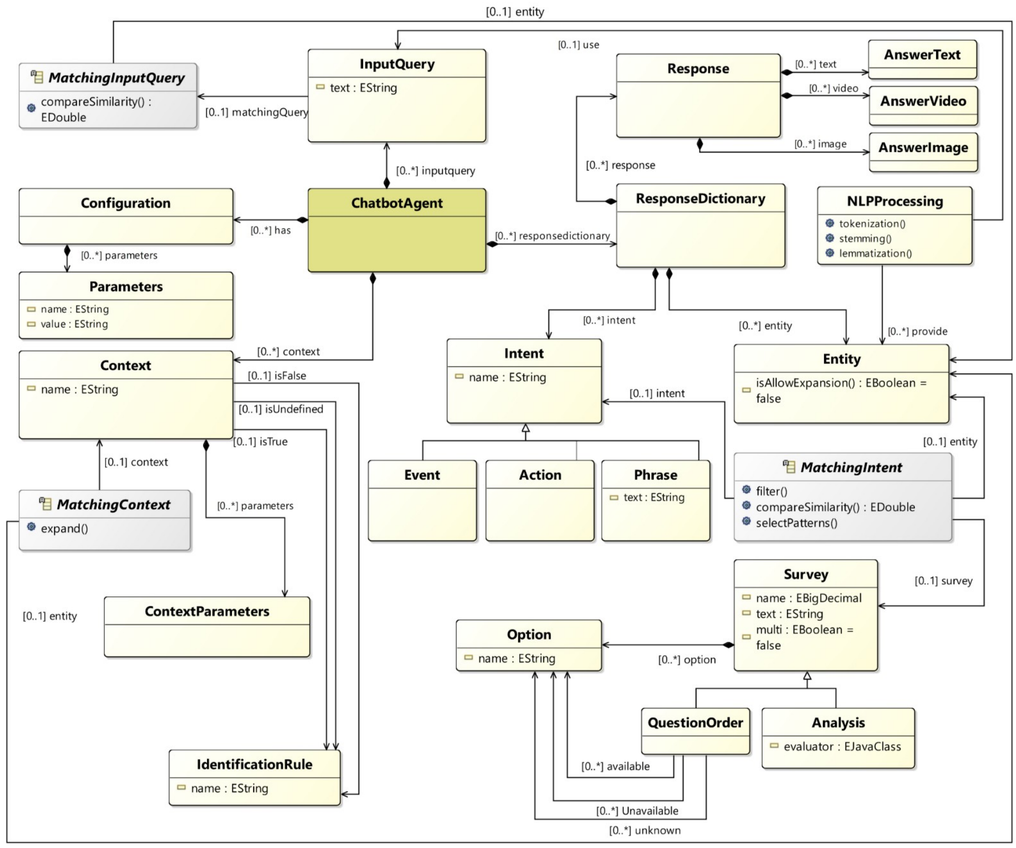


Figure 4.9 – Conceptual Organization of our Chatbot [34].

—

This technique is commonly used in natural language processing (NLP) to simplify queries or prompts without losing essential information (paraphrasing).

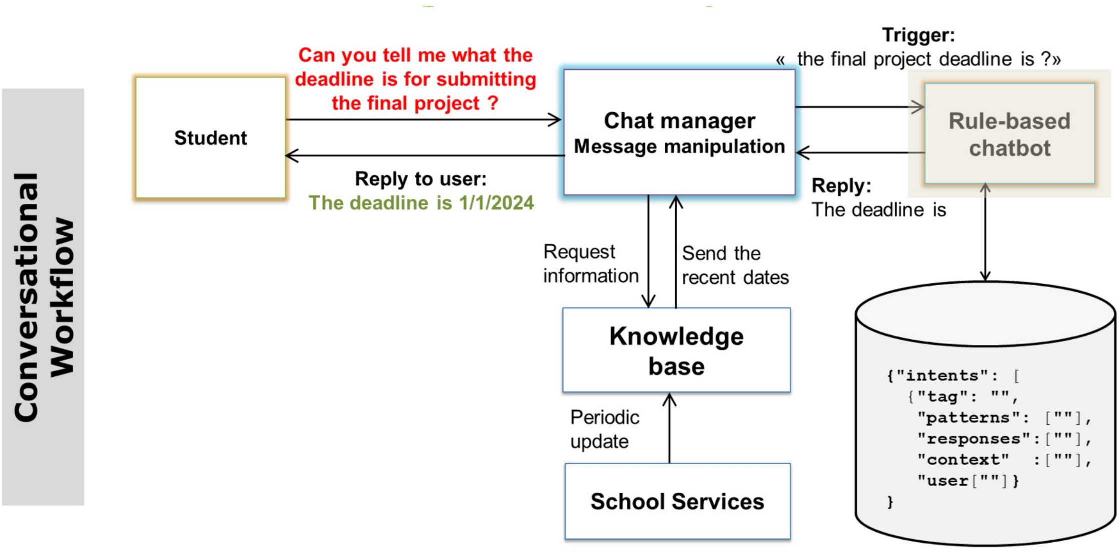


Figure 4.10 – A Context-Aware Chatbot for handling students' inquiries

4.3.5 Chatbot Manager

Our Chatbot Manager plays a key role in guiding the processing of user queries. The response can be extracted from FAQs (frequently asked questions), retrieved from a database, generated by the system, or a combination of retrieval and generation.

The Figure 4.11 shows is a block diagram of a chatbot manager. It shows the different stages a chatbot goes through to process information and respond to a user.

— **Pre-retrieval** : This stage involves indexing information, manipulating the query, and modifying data.
— **Retrieval :** This stage involves searching for and ranking information.
— **Post-retrieval :** This stage involves re-ranking information, filtering information, and enhancing information.
— **Generation :** This stage involves customizing the response for the user.

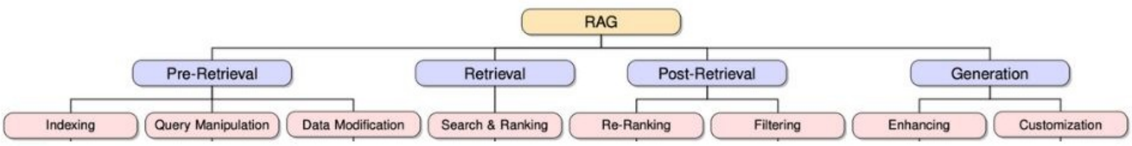


Figure 4.11 – Chatbot Manager : this components incorporates pre-retrieval steps, retrieval with search, post-retrieval steps (re-ranking, filtering).

4.3.6 Rule-based chatbot

These components utilize a response dictionary to generate a set of keywords associated with intent based on cosine similarity. To filter on intent with its corresponding response that matches correctly with the user requirement, the system uses the keywords of the response along with its associated context.As we see in Figure 4.12 :

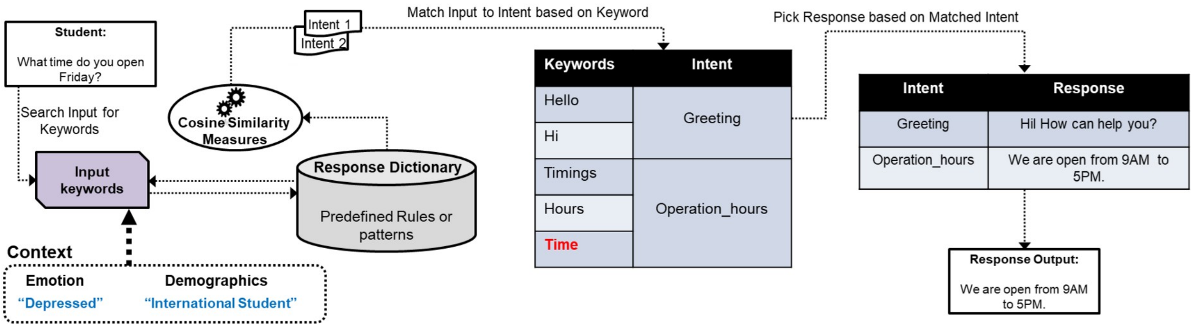


Figure 4.12 – Rule-based chatbot

Another key step is *Intent identification*. Our Chatbot looks for specific keywords to understand what actions the student wants, and classifies the inquiry accordingly, which, in turn, allows to select an appropriate response. Keyword ranking and sentence similarity are performed using n-gram, TF-IDF to convert into numerical vectors and compute the similarity with the questions stored in the RD, which are used as a training set.

$$\Box(\Box\Box, \Box) = \sum_{\Box} \Box_{\Box\Box} \times \frac{\Box_{\Box\Box}}{\Box_{\Box\Box} + \frac{\Box\,|\Box\Box|}{\Box\Box\Box\,|\Box\Box|}} \times \log\frac{|\Box|}{\Box_{\Box}} \tag{4.1}$$

This computation depends on the RD and the inquiry Q. The $\Box_{\Box\Box}$ is the term frequency of word $\Box$ in $\Box$. The summation indicates that the term's frequency has a direct impact on its importance, while the second part of the formula shows that the repetition is less important in the case of a short inquiry. The IDF component at the end gives higher weight to rare words compared to common words, where $|\Box|$ is the total number of responses, and $\Box_{\Box}$ is the number of responses containing word $\Box$.

4.3.7 Rule-based Chatbots follow pre-determined decision trees

To resolve the problem, we have utilized a "Chatbot Decision Tree." Therefore, our Rule-based Chatbots follow predetermined decision trees that we called on our model a survey. Our survey of sequence question is based on a sequence model, and follows a logical and sequential order organised as an if-then tree that is guided by the student who provides values for the parameters to select a path towards a decision (specific response), from the tree's root to a specific leaf.

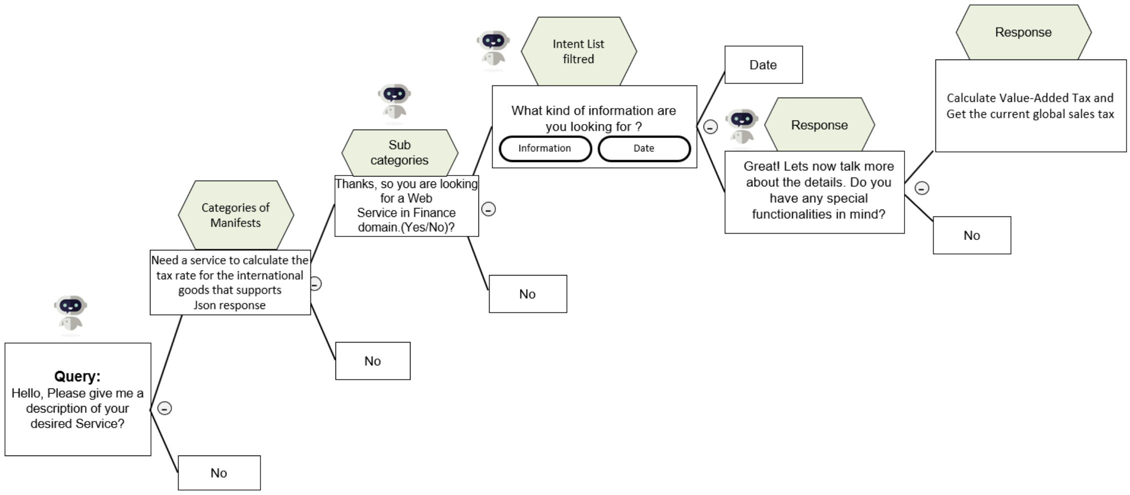


Figure 4.13 – Rule-based Chatbots follow pre-determined decision trees

4.4 Deployment architecture

Our chatbot implementation follows a rules-based approach utilizing a response dictionary stored in MongoDB. The Flask API invokes either the rule-based system or the AI model based on the user's query. Additionally, the FAQ section is dedicated to answering frequently asked questions, while the history of user interactions can be leveraged by dashboard services in the user center.

4.4.1 Overview of Flask and RESTful API

At this part of the project, we will talk about deploy our model ai using Flask as illustrated in the following figure.

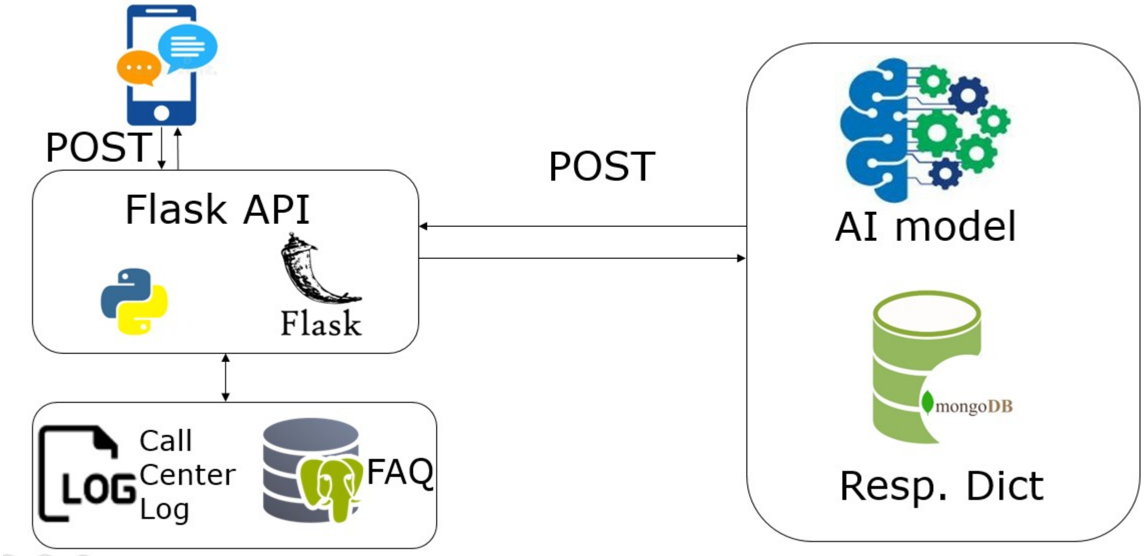


Figure 4.14 – Overview of Flask and RESTful API

4.4.2 Client-Side Interface

The client-side interface can invoke the AI model or model-based chatbot from any user interface, whether it's a tablet, mobile device, or website. This invocation is done simply by introducing the text of the query and invoking the system remotely.

4.4.3 Server-Side Application

The server-side application handles the chatbot model, whether it is an AI model or a dictionary of responses, along with the similarity metrics to generate the appropriate response.

4.4.4 Integration with External Services

Integration with external services is a technique that connects our system with the broader ecosystem. For instance, we can integrate our chatbot with Moodle LMS to explore all user interactions by using the xAPI specification (actor, verb, object). This allows our system to track and analyze user behavior, providing valuable insights and enhancing the overall functionality and user experience.

Exemple 1. ChatGPT integration throught RESTful API, *ChatGPT, offers a public RESTful API. However, there are alternative options to interact with large language models using Python.*

```python
import requests

API_KEY = "YOUR_OPENAI_API_KEY"
URL = "https://api.openai.com/v1/completions"

prompt = "What is the capital of France?"
model = "text-davinci-003"
max_tokens = 100

data = {"model": model, "prompt": prompt, "max_tokens": max_tokens}
headers = {"Authorization": f"Bearer {API_KEY}"}

response = requests.post(URL, headers=headers, json=data)

if response.ok:
  completion = response.json()["choices"][0]["text"].strip()
  print(f"ChatGPT Response: {completion}")
else:
  print(f"Error: {response.status_code}")
```

Listing 4.1 – Shortened Code for Interacting with OpenAI API

4.4.5 Performance and Optimization

Performance and optimization are essential when considering the number of users connected online, so we must think about both software and hardware solutions. Additionally, when the number of responses in the dictionary is very large, we must propose a solution to quickly exploit it using algorithms with linear or pseudo-linear complexity.

4.5 Conclusion

In higher education institutions, inquiries of a similar nature are often asked by all kinds of students, requiring timely and accurate responses that are often difficult to deliver given the number of students and the variety of inquiries. To address this issue, we proposed Yusr, an evolutive chatbot intended for students that answers common inquiries and provides dedicated, contextual answers based on a student's specific situation (academic, demographic, as well as personal goals and interests) and profile. On the other hand, Yusr also offers an aggregated view of the inquiries for educators and stakeholders, providing a dashboard that presents various views over the inquiries, requirements, and actions students operate with the chatbot.

Chapter

5

Yusr : Proof of Concept & Tooling

Contents

5.1 Introduction

Thanks to developments in artificial intelligence and natural language processing, educational chatbots like Yusr—which aim to offer individualized, interactive learning support—have been developed. Yusr enhances student engagement by providing tailored learning routes and instant feedback, hence addressing difficulties in traditional education.

To guarantee Yusr's dependability and efficacy, a proof of concept stage is incorporated into the development process. The chapter demonstrates how AI tools have the potential to revolutionize education by highlighting the creative thinking and technological know-how that went into developing Yusr.

5.2 Technology used

In this section, we present the programming environment used to develop our impenetrable system. This includes the programming language and a high-level overview of the libraries employed.

5.2.1 Programming laguage

For programming the modele of this chatbot we used Python wich is high Level General Purpose Programming language with simple syntax that allows programmer to more focus on Problem solving that on Syntax errors .

Next we will talk about the libraries we used in our implimentation.

5.2.2 Libraries used

This table represents the libraries we used to implement this project

Library	Logo	Role
Numpy	NumPy	Numerical computation
Random	pandas	Generating random number
Json	{json}	Used for loading and handling JSON data
Torch	PyTorch	Building and training neural network models
Nltk	NLTK	Natural language processing tasks

Figure 5.1 – Libraries used

- **Numpy :** It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations[17].In this project, we used library for numerical operations and handling arrays, especially for creating the bag-of-words representation.
- **Random :** The random library in Python is a built-in module that provides tools for generating random numbers and performing various randomization tasks. It's a fundamental component for applications that require an element of chance, like simulations, games, data analysis, and scientific computations[14][17].
- **Json :** The json library is a built-in module in Python that simplifies working with JSON data (JavaScript Object Notation). It provides functionalities for both encoding Python data structures into JSON format and decoding JSON strings back into Python objects[13][38]. in this project we used it to load and parse the intents.json file, which contains the training data for the chatbot.
- **PyTorch :** PyTorch, often referred to simply as Torch, is a popular open-source machine learning library built for Python. It's particularly well-suited for deep learning applications [**softwarepytorch**]. We used this pytorch library for building and training the neural network model and provides tensor computation, automatic differentiation, and GPU support.
- **Nltk :** NLTK,or Natural Language Toolkit, is a popular open-source Python library used for Natural Language Processing (NLP) tasks[6]. in our project this library used to Provides tools for tokenization, stemming, and accessing synonyms from WordNet.

5.3 Our Implmentation

In this section, we have divided our source code into three modules : (i) the model, (ii) the apprentissage, and (iii) the test to determine the fracture interface. We will present some of each module's directives.

5.3.1 Data Set

The example base is organized as "intent, pattern and tags" . This data is stored in a JSON file and can be leveraged by pre-processing and training modules. The example below illustrates the structure of a record in the example base, which keeps track of the training examples.

```
{
  "intents" : [
    {
      "tag" : "greeting",
      "patterns" : [
        "Hi",
        "Hey",
        "How are you",
        "Is anyone there ?",
        "Hello",
        "Good day"
      ],
      "responses" : [
        "Hey there ! How can I assist you today ?",
        "Hello ! Welcome to our service. How may I help you ?",
        "Hi ! How are you doing ? What can I do for you ?",
        "Hello ! Thanks for reaching out. What do you need assistance with ?"
      ]
    },
    {
      "tag" : "goodbye",
      "patterns" : ["Bye", "See you later", "Goodbye"],
      "responses" : [
        "Goodbye ! If you need anything else, feel free to ask.",
        "See you later ! Have a great day !",
        "Bye ! Take care and come back soon."
      ]
    },
    ...
    }
```

Listing 5.1 – An excerpt of JSON code corresponding to the dataset

5.3.1.1 Data augmentation

Data augmentation is the process of generating new data examples for training a model. Data augmentation is a useful technique for providing more information from less data[21].

Figure 5.3 – Data Processing Workflow (Subfigures)

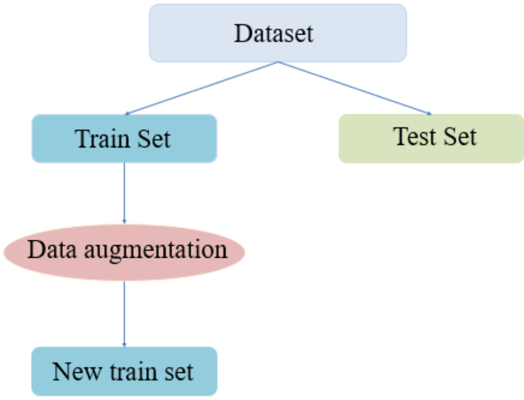


Figure 5.2 – Data augmentation

To make our dataset larger, we used data augmentation techniques to create additional data from the data we had. These technologies are :

- **Synonym Replacement :** replace words with thier synonyms.
- **Random Insertion :** Find a synonym for a random word in the sentence that is not a stop word and replace it with the word.
- **Random Swap :** Randomly choose two words from the sentence and flip their positions.
- **Random Deletion :** Choose a random word and remove it from the sentence.

To make our dataset larger we used the techniques of data augmentation to generate additional data from the data we have.

Exemple 1. Example of data augmentation *Consider the following question :*

How do I find the course schedule for next semester ?

We will now generate augmented versions of this question using various data augmentation techniques.

- Q1 : Where can I see the course schedule for next semester ?
- Q2 :How can I access the course schedule for next semester ?
- Q3 :Comment puis-je trouver l'emploi du temps des cours pour le prochain semestre ?
- Q4 : ¿Cómo puedo encontrar el horario de cursos para el próximo semestre ?
- Q5 : In which section of the website can I find the course schedule for next semester ?
- Q6 : What steps do I need to follow to find the course schedule for next semester ?

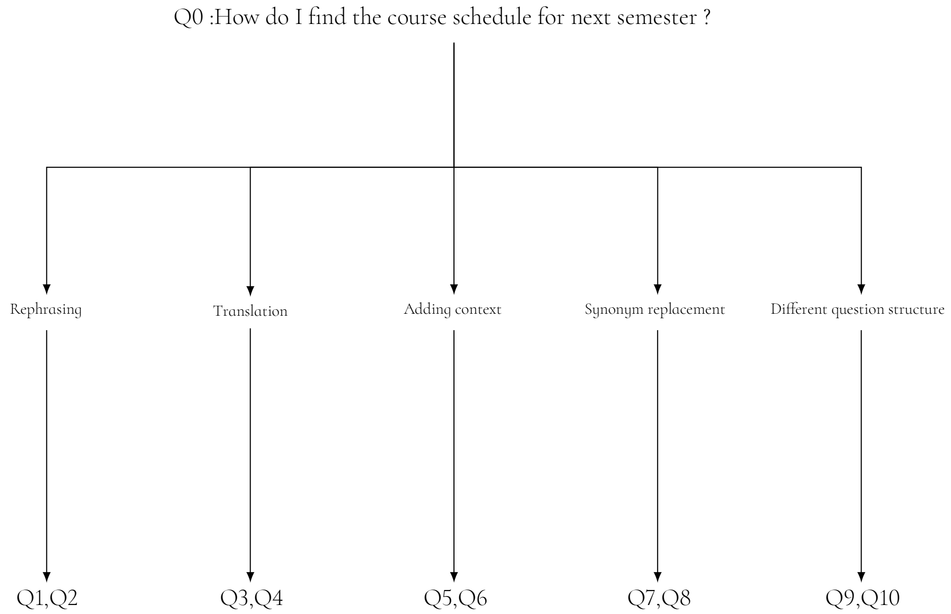


Figure 5.4 – Tree Diagram of Course Schedule Inquiry Variations

- Q7 : How do I locate the course schedule for next semester ?
- Q8 : Where do I find the timetable for courses next semester ?
- Q9 : The course schedule for next semester is found where ?
- Q10 : Tell me how to find the course schedule for next semester

5.3.2 Model

The Listing 5.2 shows the code of our AI model **"model.py"** :

```python
import torch
import torch.nn as nn


class NeuralNet(nn.Module) :
    def __init__(self, input_size, hidden_size, num_classes) :
        super(NeuralNet, self). __init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x) :
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out
```

Listing 5.2 – An excerpt of Python code corresponding to the AI model

The NeuralNet class is a simple feedforward neural network in PyTorch. It consists of :

— `Layers` : Three fully connected layers (*l1*, *l2*, *l3*).

—

— `Activation` : ReLU activation function after the first and second layers.

The *forward method* passes input x through these layers sequentially, applying *ReLU* activation after the first and second layers. The final layer output is returned without any activation, suitable for further processing like applying *softmax* during loss computation.

TheListing 5.3 shows the code of**"nltk_utils.py"** which is code of prepossessing steps

```
...
def tokenize(sentence) :
    """
    split sentence into array of words/tokens
    a token can be a word or punctuation character, or number
    """
    return nltk.word _tokenize(sentence)


def stem(word) :
    """
    stemming = find the root form of the word
    examples :
    words = ["organize", "organizes", "organizing"]
    words = [stem(w) for w in words]
    -> ["organ", "organ", "organ"]
    """
    return stemmer.stem(word.lower())
    ...
```

Listing 5.3 – An excerpt of Python code corresponding to the preprossessing steps

This code defines functions using NLTK for text processing :

tokenize(sentence) :

Splits a sentence into words or tokens using nltk.word_tokenize.

stem(word) :

Converts a word to its root form using a stemmer's stem method after converting the word to lowercase.

TheListing 5.4 shows the code of**"chat.py"** which is a part of chatbot implementation we used the pre-training model "model.py"

```
...
def get_response(msg) :
    sentence = tokenize(msg)
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)

    output = model(X)
    _, predicted = torch.max(output, dim=1)

    tag = tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]
    if prob.item() > 0.75 :
        for intent in intents['intents'] :
            if tag == intent["tag"] :
                return random.choice(intent['responses'])

    return "I do not understand..."
```

Listing 5.4 – An excerpt of Python code corresponding to the chatbot code

The pre-trained neural network model and related data are loaded by this script, which then uses user input to forecast purpose and, to a certain extent, produces a suitable response based on that intent. It returns a fallback message stating that the input was not comprehended if the confidence level is low.

5.3.3 Model learning

Their we have the code structure of training the model then save it in the file **"data.pth"** TheListing 5.8 shows the code we use it to train our model :

```
...
for epoch in range(num _epochs) :
    for (words, labels) in train _loader :
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero _grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0 :
        print (f'Epoch [{epoch+1}/{num _epochs}], Loss : {loss.item() :.4f}')

print(f'final loss : {loss.item() :.4f}')
...
```

Listing 5.5 – An excerpt of Python code corresponding to the training of the model

This code snippet trains a PyTorch model over multiple epochs :

1. *Epoch Loop* : Runs for num_epochs iterations.
2. Batch Loop : Iterates over batches from train_loader.
3. *Data Preparation* : Moves words and labels to the device, with labels cast to long.
4. *Forward Pass* : Computes model outputs from words.
5. *Loss Calculation* : Calculates loss between outputs and labels.
6. *Backward Pass & Optimization* :
7. Zeroes gradients.
8. Computes gradients.
9. Updates model parameters.
10. Progress Reporting : Prints loss every 100 epochs.
11. *Final Loss* : Prints the final loss after training.

This loop effectively trains the model, updates its parameters, and provides progress updates.

5.3.4 Model Testing

We do some tests to evaluate the effectiveness of our chatbot.

5.3.4.1 Intent identification testing

In this experiment, we evaluated the accuracy of our chatbot, YUSR, by testing its performance in identifying user intents across three categories : Simple, Intermediate, and Complex. Each category represents a varying level of question complexity commonly encountered by users interacting with the chatbot. We conducted manual tests where a set of predefined questions were presented to the chatbot, and its responses were compared against the expected answers.

Additionally, we observed the influence of dataset size on the performance and accuracy of our chatbot. As the dataset size increases, the chatbot's ability to accurately identify user intents may be affected. We analyzed this impact on YUSR's performance to understand how scalability of data influences its accuracy in responding to user queries."

Test categories	Identification rate (%)
Simple	85
Intermediate	70
Complex	50

Table 5.1 – Intents Tests

We notice from this table that our chatbot arrives has identified the intent with a precision rate/This result is acceptable in the context of higher education, which is shown by the number of very important students. What is minimizes the burden of admin and technicians.

5.3.4.2 Response Identification Testing

To test our chatbot YUSR, we conduct a series of manual tests. For each test, we verify if our chatbot has identified the correct response. The tests are categorized into three categories of questions based on their complexity. The following table shows the question and answer identification results :

Category	Identification Rate	Correct Response Rate
Simple	95%	88%
Intermediate	80%	72%
Complex	65%	55%

Table 5.2 – Q|A Identification Results

Category	**Question**	**Expected Answer**	**Chatbot's Response**	**Correct ? (Yes/No)**
Simple	When is the final exam for the Mathematics course ?	June 15	June 15	Yes
Simple	What is the passing grade for the Physics exam ?	50 Intermediate	How many credits is the Chemistry lab course worth ?	3 credits
3 credits	Yes			
Intermediate	What is the format of the History exam ?	Multiple choice and essay	Multiple choice and essay	Yes
Complex	Can you explain the grading criteria for the Computer Science project ?	Based on functionality, code quality, and documentation	Based on functionality, code quality, and documentation	Yes
Complex	What are the main topics covered in the final exam for Organic Chemistry ?	Reaction mechanisms, spectroscopy, and synthesis	Reaction mechanisms, spectroscopy, and synthesis	Yes
Simple	Where can I find the schedule for final exams ?	On the university website	On the university website	Yes
Intermediate	What are the prerequisites for enrolling in the Advanced Calculus exam ?	Completion of Calculus I and II	Completion of Calculus I and II	Yes
Complex	How should I prepare for the final exam in Econometrics ?	Review lecture notes, solve past papers, and understand key concepts	Review lecture notes, solve past papers, and understand key concepts	Yes
Complex	What is the policy for requesting a regrade on an exam ?	Submit a formal request within two weeks of receiving the grade	Submit a formal request within two weeks of receiving the grade	Yes

Table 5.3 – Q|A Identification Results for Higher Education Exam Queries

5.4 UI/UX of our assistant Tool

In this section, we will explain the UI/UX design process used to develop our assistant tool.

5.4.1 Technology used

TO create our mobile application we used the following technology : For the UI/UX design we used FIGMA : Figma is a vector-based tool that lives in the cloud, allowing users to work anywhere from a browser. It is a design and prototyping tool designed for digital designers and eased enough to be used by non-specialists while sharing or extracting files.

For the mobile application we used Flutter in Android Studio : Flutter is an open-source UI software development kit created by Google. It can be used to develop cross platform applications from a single codebase for the web, Fuchsia, Android, iOS, Linux, macOS, . . .

For the data base we usede My SQL : MySQL is an open-source relational database management system that stores, manages, and retrieves structured data using Structured Query Language (SQL).

5.4.2 Mobile app design process

In designing this application, we followed a set of stages. The Figure 5.5 illustrates the Mobile app design process.

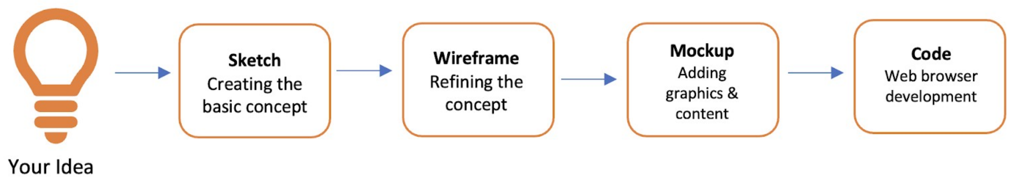


Figure 5.5 – Mobile app design process

5.4.2.1 Sketching

The term "sketching" in app design describes the preliminary phase of producing crude, low-fidelity sketches to gather inspiration and layout ideas for your mobile application [10]. In light of the team's discussions, we have developed this preliminary design.

The Figure 5.6 shows the sketching of our mobile application during brainstorming with our supervisor

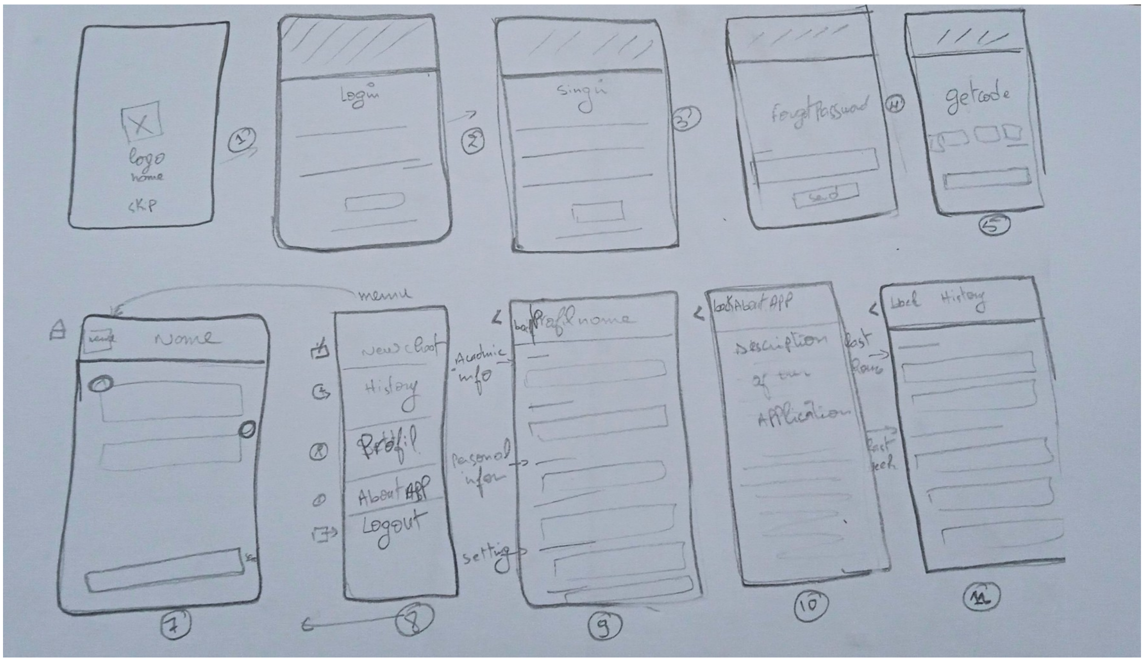


Figure 5.6 – Sketching of our mobile application during brainstorming with our supervisor.

5.4.2.2 Wireframing

Wireframes are a fundamental tool in the application mobile design process, serving as simplified visual representations of a applicatoin's layout and functionality [30].

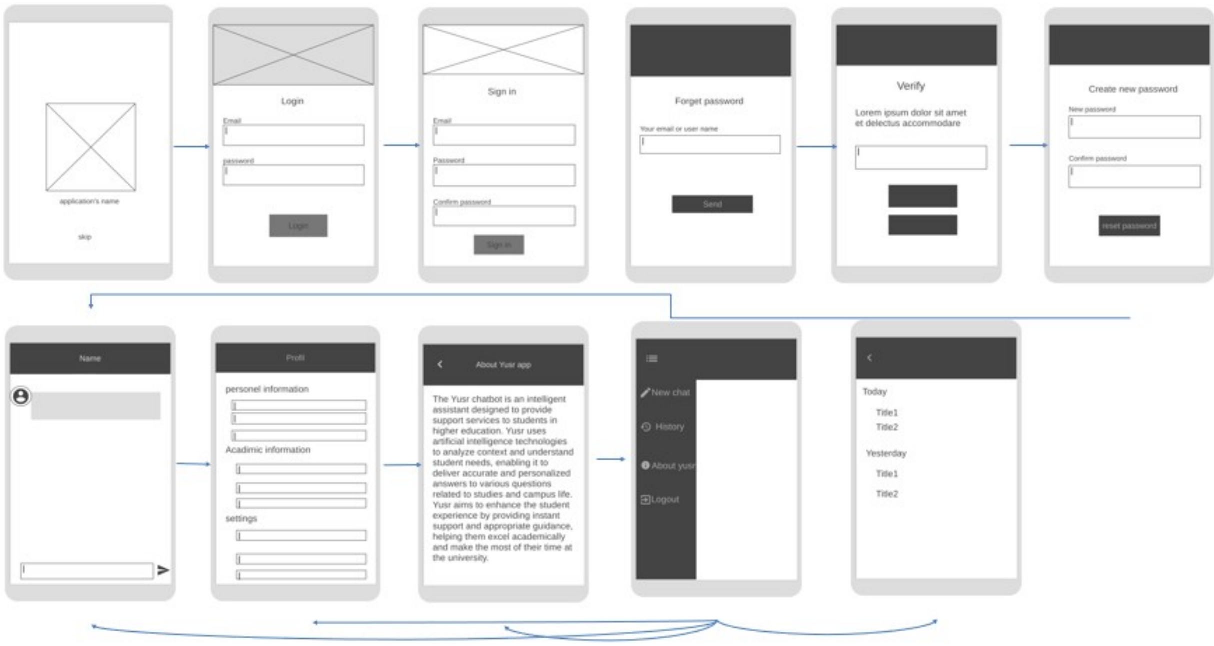


Figure 5.7 – Wireframe of our application

5.4.2.3 Storyboards

A storyboard acts as a visual blueprint, laying out the key ideas and structure of a planned experience. It utilizes a sequence of illustrations, images, or screens to provide a pre-visualization and establish the order of events. This allows for early feedback and refinement before the actual experience is created.

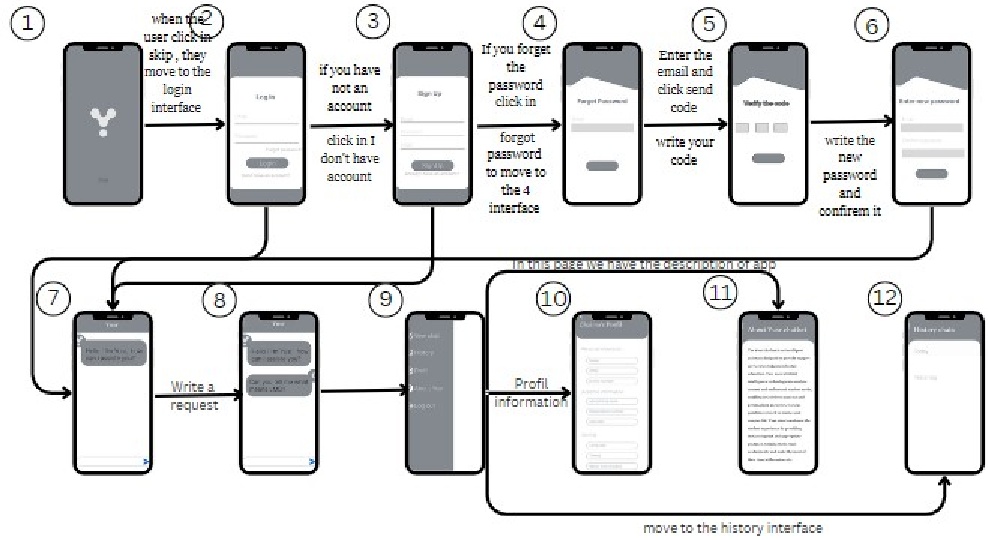


Figure 5.8 – Storyboards of our application

5.4.2.4 Tool Snapshots

In this section we will define the interface of our mobile application

Splash screen

This is the Splash screen the first screen contains the logo.

Sign_UP&Log in

The Sign_up screen where the user add their personal information to create account.

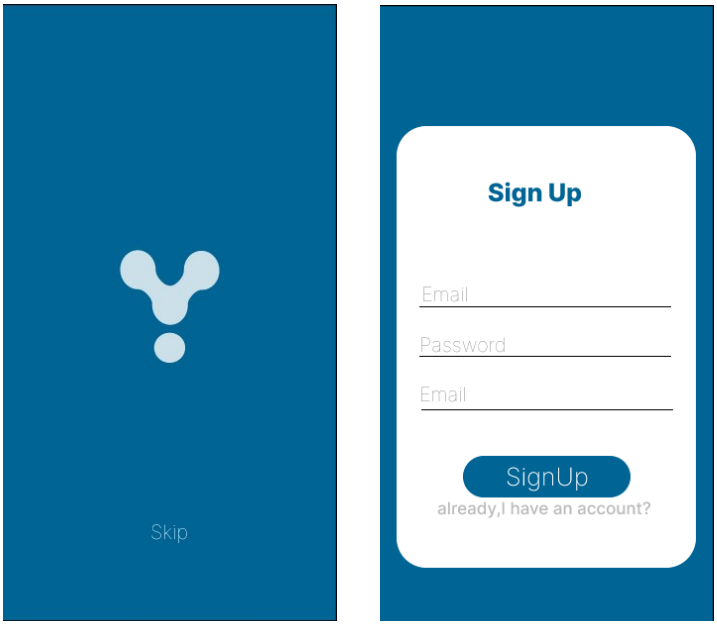
The Login screen from where the user log in their account

Menu Screen

The Menu screen where are the menu of the app, where user can add new chat or show their previous chats, modifie the profile information or get information about the bot.

Chat Screens

This screens are examples of a conversation between Yusr and user.



(a) brand of your mobile application. (b) User Interface User Sign Up

Figure 5.9 – `Yusr` main GUI and its component module panels (A).

Profile Screen

The profile screen can change their information and add their academic information.

About Yusr Screen

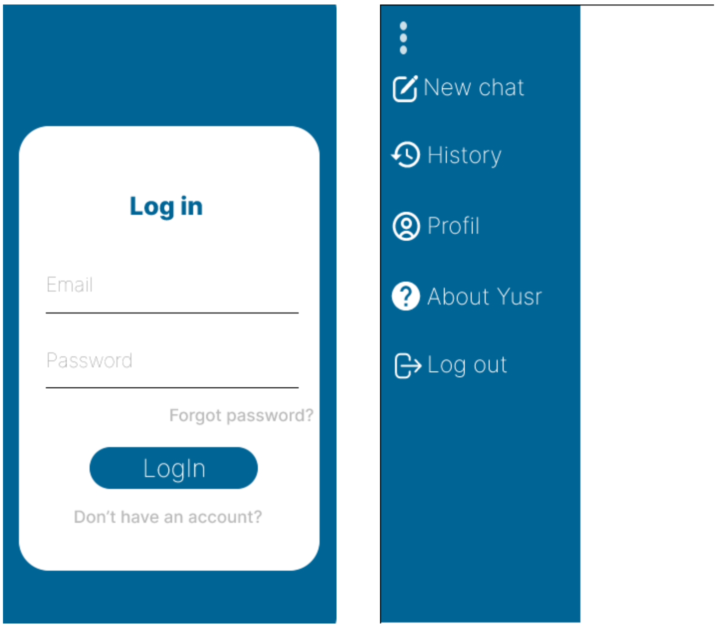
This screen have information about our bot for helping the user.

5.5 Deployment Architecture

In this section, we present how to deploy our AI chatbot model generated by the NLP process to the backend.

5.5.1 Component of the Deployment Architecture

The diagram illustrates a chatbot system where a user interacts with a mobile phone or laptop application. This application communicates with a Flask API, which acts as a web server. The Flask API utilizes an AI model in the backend to process and respond to user queries. Here,



(a) User Interface User Login in

(b) The Interface Menu of our application

Figure 5.10 – Yusr main GUI and its component module panels (B).

the text "jQuery AJAX" likely refers to the method the mobile application uses to communicate with the Flask API.

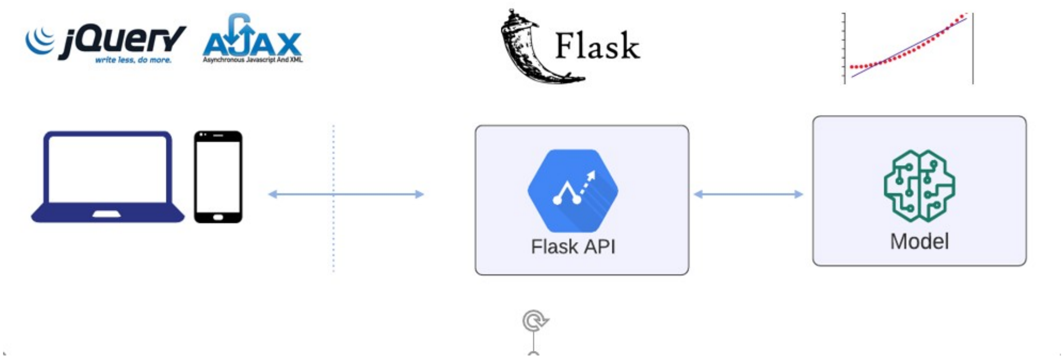
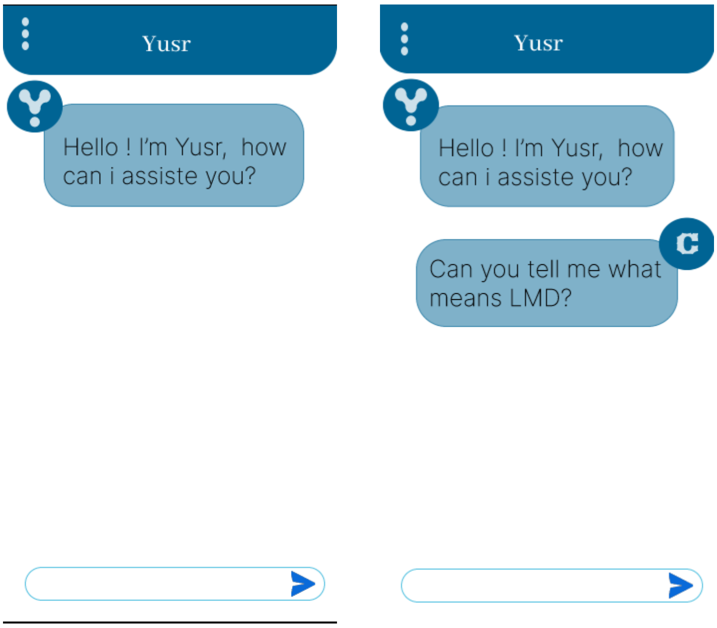


Figure 5.13 – Deployment Architecture

Here's a breakdown of the components :

— **User Interface :** This represents the mobile phone or laptop application that users interact with to initiate chats with the chatbot.



(a) User Interface Chat screen. (b) User Interface Chat screen2

Figure 5.11 – Yusr main GUI and its component module panels (A).

- **Flask API :** This is a web server built using the Flask framework in Python. It acts as an intermediary between the user interface and the AI model.
- **AI Model :** This represents the artificial intelligence model trained for chatbot conversations. It resides in the backend and is responsible for processing and generating responses to user queries.

5.5.2 Step of AI Model Deployment

Step of AI Model Deployment ocntain three main steps Save the AI Model, Load the AI Model and Invoke the AI ModeL ;

5.5.2.1 Save the AI Model

This step saves the trained model's structure and data into a file, so you can use it later. The way you do this depends on the machine learning library you used, like TensorFlow, PyTorch, or spaCy.

(a) User Interface Profile screen2 (b) The Interface About Yusr Chatbot

Figure 5.12 – `Yusr` main GUI and its component module panels (B).

```
from tensorflow.keras.models import model

# Assuming your trained model is assigned to 'model' variable
model.save('catbot _model.h5')   # Replace 'catbot _model.h5' with your desired filename
```

Listing 5.6 – An excerpt of saving the AI Mode

5.5.2.2 Load the AI Model

When you want to use the model, you need to load it from the saved file. To do this, use the right function from your chosen library to read the model data and create a working version in memory.

```
from tensorflow.keras.models import load   _model

# Replace 'catbot _model.h5' with the path to your saved model
model = load _model('catbot _model.h5')
```

Listing 5.7 – An excerpt of pyhton for Loading the AI Model

5.5.2.3 Invoke the AI Model

After loading the model, you can use it to make predictions on new data. You need to give the model input data that is prepared the same way as during training. The model will then produce an output, like a classification, a text, or another prediction, depending on what it was designed to do.

```
from flask import Flask, request, jsonify
import your_model_library  # Replace with your NLP library (e.g., spacy, transformers)

# Initialize Flask app
app = Flask(__name__)

# Load your AI model
model = your_model_library.load('path/to/your/model')    # Replace with actual model path

def predict(text) :
  """
  This function takes user input text, preprocesses it, and makes predictions using your model.
  """
  # Preprocess the text (e.g., cleaning, tokenization) based on your model's requirements
  processed_text = preprocess_text(text)   # Replace with your preprocessing function

  # Make predictions using your model
  prediction = model(processed_text)

  # Process the prediction for the response (e.g., converting probabilities to classes)
  return process_prediction(prediction)   # Replace with your prediction processing function

@app.route('/predict', methods=['POST'])
def make_prediction() :
  # Get user input text from the request
  data = request.get_json()
  if not data or 'text' not in data :
    return jsonify({'error' : 'Please provide text data in the request'}), 400

  text = data['text']

  # Make predictions using the predict function
  prediction = predict(text)

  return jsonify({'prediction' : prediction})

if __name__ == '__main__' :
  app.run(debug=True)
```

Listing 5.8 – An excerpt of pyhton for invoking the AI Model

5.6 Conclusion

In this chapter, we present the implementation of our AI model using Python and a set of libraries. We also describe the model testing process using various scenarios. The results demonstrate that our model can effectively assist administrators, technicians, and students 24/7. Finally, we showcase our mobile application and the deployment architecture of the model utilizing the Flask framework.

Chapter

6

Conclusion and Perspectives

Contents

6.1 Conclusion

Nowadays, we can find Chatbots for student assistance services. While these tools reduce costs, and their presence is percolating to a wide range of areas such as education, the students' context can become obscured under several dimensions. Moreover, selecting the most appropriate response by a Chatbot requires to perceive all the context that cover the students question or requirements, consider users' personalized related preferences and awareness of the user profile at varied spatial granularities. For providing efficient student assistance towards a context-aware Chatbot, bridging the models of the user profile, context and intent becomes a necessity. Such an explicit connection is an important prerequisite for designing a Chatbot model for the assistance services domain in order to contextually satisfy user queries concerning intent recommendations. To alleviate these problems, we have proposed a context-aware Chatbot for student assistance services that includes a students context with its dimensions (i.e. Personal information, Demographics information, Academic information and outcome, Emotion). Our approach and has been implemented as support solution to assist students and support staff, its supports both students queries response and support staff by the Chatbot dashboard.

6.2 Perspectives

In the future, we plan to extend our framework with NLP techniques for emotion detection, also in order to extract other information about students from for example forum discussion in learning management systems. We are currently improving our editor of the Chatbot to a user-friendly interface, and later we plan to perform a user study with UX designers to assess the advantages of our approach. Finally, we plan to evaluate how our Chatbot can significantly increase efficiency and effectiveness of the students' questions in higher education.

Chapter

7 Project Management

Contents

7.1 Introduction

Any project must begin with a planning phase that tries to specify the tasks that need to be completed, manage risks, and provide updates on the project's status. We describe in this chapter the project management approach we used. We first go into detail about our organization. Next, we provide the overall work planning. After that, we talk about the risk analysis.

7.2 Project planning and organization

Planning is a necessary step before any project can begin. Its goals are to specify the activities that must be completed, manage risks, and provide updates on the project's status. We've created a thorough management strategy to make sure everything goes according to plan and on schedule for our project.

7.2.1 Organization

Our project is broken down into several milestones. In order to carry out the work of each milestone as well as possible, we have based ourselves on a working method that requires weekly or even daily contact between the students and the supervisor. This contact is reflected in the different types of meetings.

Contact Types	Duration	Participant	Objective
Emails	-	Students & Supervisor	Update on the progress of the work
Discussion meetings	1h - 2h	Students & Supervisor	Solving problems
Work presentation meetings	1h	Students & Supervisor	Presentation of work done

Table 7.1 – Contact types & meetings

7.2.2 Planning

In order to better illustrate the planning of the tasks we carried out within this project, we propose the Gantt diagram . This diagram shows the chronological sequence and detail of milestones completed.

Milestone	Description	Duration per day
1	Search for information	20
2	Analysis of needs	10
3	Design	7
4	Code structure	5
5	Implementation	15
6	Train & test model	10
7	UI/UX design	8
8	Deployement	7
9	Application test	8

Table 7.2 – List of tasks

7.2.3 Risk assessment

On a project, we are all led to encounter one or more risks that can prevent its progress on time. It is for this reason that we have previously assessed a set of risks that may arise in the context of this project.

Number	Name of risk	Category
1	Absence of the supervisor	human
2	Absence of students	technical
3	Lack or unavailability of useful information	technical

Table 7.3 – List of riskes

Absence of the supervisor :

The absence of the supervisor can cause problems for the rest of the project, because students may indeed have difficulties with : - The validation of his work, to be able to move on to another task. - A lack of work leads.

Absence of students :

A student is brought at any time to be sick, or have situations that force him to be absent.

7.3 Conclusion

In this chapter, we have shown that we are able to assess risks in order to overcome them if they are proven. The planning has also shown its advantages by always having a benchmark of the progress of the internship and by making it possible to meet the deadlines of the planned tasks.

Part V

Annexes

البطاقة التقنية للمشروع

Boulefred Yassmina Boumaza Chaimaa	الاسم و اللقب Votre prénom et nom Your first and last Name
Yusr-Bot	الاسم التجاري للمشروع Intitulé de votre projet Title of your Project
	الصفة القانونية للمشروع Votre statut juridique Your legal status
6089303760 6089303397	رقم الهاتق Votre numéro de téléphone Your phone number
boulefredyassmina@gmail.com boumazachaimaa32@gmail.com	البريد الالكتروني Votre adresse e-mail Your email address
Tiaret-Tiaret	مقر مزاولة النشاط (الولاية- البلدية) Votre ville ou commune d'activité Your city or municipality of activity

طبيعة المشروع

A mobile application that help the users in their daily live, answer their questions according to their context to know their intent. In our case, the users are the students in the higher education.	

القيمة المقترحة أو العرض المقدّم Value Proposition

تحديد المشكل الذي يواجهه الزبون

The charge of students queries for the vice dine and the administrations	ما هي المشكلة التي تريد حلها؟
Market research reveals that this problem is present throughout the year	ما هي البيانات المتوفرة لديك التي تدل على وجود المشكلة المحددة؟
	ما هي المشاريع الأخرى التي استهدفت نفس المشكلة والتي جرى تنفيذها؟
listen to students 24/7	ماهي أهداف مشروعك و/أو نتائجه المتوقعة؟

القيمة المقترحة وفق المعايير التالية

	القيمة المبتكرة أو الجديدة
	القيمة بالتخصيص
	القيمة بالسعر

Design a simple and intuitive to ensure a smooth experience for all ages. - Use attractive design and colors to improve the user experience.	القيمة بالتصميم
Develop a stable and unstable application bugs to ensure a user experience smooth and without technical problems. - Ensure a quick response to requests users.	القيمة بالأداء العالي
Provide available technical support 24/7 to help in case of emergency. - Offering excellent, responsive and to solve any problems and Answer users' questions.	القيمة بالخدمة الشاملة
	القيمة المبتكرة أو الجديدة
	قيم أخرى

Customer Segments شرائح العملاء أو الزبائن

Géographique الجغرافية	Démographique (B2C)	Démographique (B2B)	Psychographique العوامل النفسية و الشخصية	Comportemental السلوكيات
Continent القارة Africa	Age العمر 81+	Secteur القطاع Services sector	Classe sociale لطبقة الاجتماعية Middle class Lower class	Usage استخدام
Pays الدولة Algeria	Sexe الجنس Female Mal	Nombre d'employés عدد العمال في القطاع	Niveau de vie المستوى المعيشي Standard of living Medium Weak high	Loyauté الوفاء
Région الجهة West	Revenus annuel متوسط الدخل	Maturité de l'entreprise نضج المؤسسة	Valeurs القيم Accessibility Security User-friendliness Economy durability	Intérêt اهتمام
Département الولاية Tiaret	Etat matrimonial الحالة الاجتماعية	Situation financière الحالة المالية للمؤسسة	Personnalité الشخصية Reliability. Social responsibility. Accessibility. Community involvement.	Passion الهواية و شغف
llVille الدائرة او البلدية	Niveau d'étude المستوى الدراسي	Détention/ actionnariat	Convictions المعتقدات	Sensibilité حساسيات

Tiaret	All	الملكية/المساهمة		
Quartier الحي /	Profession المهنة All	Valorisation/ capitalisation boursière التقييم / القيمة السوقية	Présence digitale et sur les réseaux sociaux استعمال التكنولوجيا في التواصل Possibility of Communication with customers on Networks Social	Habitude de consommation عادة الاستهلاك
Climat المناخ /	Culture الثقافة /	Business model نموذج الأعمال	Centres d'intérêts مراكز الاهتمام Community and social connection. Financial Economics.	Mode de paiement طرق الدفع
	Religion الدين Islam	Secteur servi القطاع الذي يخدمه		Connaissance المعرفة
	Langue اللغة Arabic French English	Technologie utilisée التكنولوجيا المستعملة Flutter Python		Nature de la demande طبيعة الطلب

		Mysql		
		Format du produit ou packaging شكل المنتج أو التعبئة والتغليف Mobile app		Fréquence d'achat عدد مرات الطلب على السلعة

Channels قنوات التوزيع

Sell services through an app Mobile and web app.	المبيعات المباشرة
/	تجار الجملة
	الموزعون
/	توزيع التجزئة

Customer Relationship العلاقة مع العملاء

Service client. Collect customer information. Analyze data. Deliver a customer experience Custom.	كيف تدير علاقاتك مع العملاء؟
	ماهية أهم البرامج التي ستعتمد عليها في ادارة العلاقة مع الزبون Microsoft Dynamics Monday CRM

	Zoho CRM الخ...............................

الشركاء الأساسيون Key Partners

الشركاء	معلومات حول الشركاء	طبيعة الشراكة
الشريك الأول Universities		
الشريك الثاني **Academic institutions**		الشريك ليس منافس و انما هي عملية ترتيب متبادل المنفعة يكون لنا معه مصلحة مشتركة في تطوير منتجات جديدة إستراتيجية مصممة لتقليل المخاطر ، والتي قد ترتبط بإحضار منتج جديد إلى السوق علاقتنا مع الشريك هي علاقات بين المشتري والمورد
الشريك الثالث		الشريك ليس منافس و انما هي عملية ترتيب متبادل المنفعة يكون لنا معه مصلحة مشتركة في تطوير منتجات جديدة إستراتيجية مصممة لتقليل المخاطر ، والتي قد ترتبط بإحضار منتج جديد إلى السوق علاقتنا مع الشريك هي علاقات بين المشتري والمورد

قم بكتابة قائمة الشركاء الرئيسيون لمشروعك بالتفصيل مع ذكر الإسم، الهاتف، العنوان...إلخ

هيكل التكاليف structure Costs

تكاليف التعريف بالمنتج أو المؤسسة Frais d'établissement	From 100.000da -Marketing on social networks (Facebook, Instagram). - Content marketing (creation of videos). -Google advertising.

Our project depends only on electricity. From 30.000da	تكاليف الحصول على العدادات (الماء- الكهرباء)....... Frais d'ouverture de compteurs (eaux-gaz-….)
/	تكاليف)التكوين- برامج الاعلام الالي المختصة(Logiciels, formations
From 20.000 da Mark registration	Dépôt marque, brevet, modèle تكاليف براءة الاختراع و الحماية الصناعية و التجارية
From100.000 da	Droits d'entrée تكاليف الحصول على تكنولوجيا او ترخيص استعمالها
/	Achat fonds de commerce ou parts شراء الأصول التجارية أو الأسهم
/	Droit au bail الحق في الإيجار
/	Caution ou dépôt de garantie وديعة أو وديعة تأمين
From 20.000 da	Frais de dossier رسوم إيداع الملفات
From 50.000 da	Frais de notaire ou d'avocat تكاليف الموثق-المحامي-.........
Frome 50.000 da	Enseigne et éléments de communication تكاليف التعريف بالعلامة و تكاليف قنوات الاتصال
/	Achat immobilier شراء العقارات
From 200.000 da	Travaux et aménagements الأعمال و التحسينات الاماكن
From 800.000 da - Two Desktops for admins. -Server for hosting.	Matériel الآلات- المركبات- الاجهزة
From 300.000 da Tables, chairs, printers,	Matériel de bureau تجهيزات المكتب
Hosting: Cloud / Algeria Telecom From 500.000 da	Stock de matières et produits تكاليف التخزين
From 1.000.000 da	Trésorerie de départ التدفق النقدي (الصندوق(الذي تحتاجه في بداية المشروع.

المجموع =3.500.000 da

نفقاتك أو التكاليف الثابتة الخاصة بمشروعك

From 50.000 da	Assurances التأمينات
Internet Pack 30 MB 24999 DA/month	Téléphone, internet الهاتف و الانترنت
/	Autres abonnements اشتراكات أخرى
/	Carburant, transports الوقود و تكاليف النقل
/	Frais de déplacement et hébergement تكاليف التنقل و المبيت
From 30.000 da	Eau, électricité, gaz فواتير الماء – الكهرباء- الغاز
/	Mutuelle التعاضدية الاجتماعية
From30.000 da	Fournitures diverses لوازم متنوعة
/	Entretien matériel et vêtements صيانة المعدات والملابس
From 20.000 da	Nettoyage des locaux تنظيف المباني
From 50.000 da	Budget publicité et communication ميزانية الإعلان والاتصالات

المجموع =300.000 da

مصادر الإيرادات Revenue Stream

2.000.000 da	Apport personnel ou familial المساهمة الشخصية أو العائلية
/	Apports en nature (en valeur) التبرعات العينية
/	Prêt n°1 (nom de la banque) قرض رقم 1 اسم البنك
/	Prêt n°2 (nom de la banque) قرض رقم 2 اسم البنك

/	Prêt n°3 (nom de la banque) قرض رقم3 اسم البنك
/	Subvention n°1 (libellé) منحة 1
/	Subvention n°2 (libellé) منحة 2
/	Autre financement (libellé) تمويل آخر

المجموع =2.000.000 da

رقم الأعمال

Votre chiffre d'affaires de la première année بيع المنتج في السنة الأولى

متوسط أيام العمل في الشهر	بيع المنتج في السنة الأولى
20	1Mois الشهر
20	2Mois الشهر
20	3Mois الشهر
20	4Mois الشهر
20	5Mois الشهر
20	6Mois الشهر
20	7Mois الشهر
20	8Mois الشهر
20	9Mois الشهر
20	10Mois الشهر
20	11Mois الشهر
20	12Mois الشهر

المجموع =
النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الأولى؟

Votre chiffre d'affaires de la deuxième année بيع المنتج في السنة الثانية

متوسط أيام العمل في الشهر	بيع المنتج في السنة الثانية
20	1Mois الشهر
20	2Mois الشهر
20	3Mois الشهر

الشهر 4Mois	20
الشهر 5Mois	20
الشهر 6Mois	20
الشهر 7Mois	20
الشهر 8Mois	20
الشهر 9Mois	20
الشهر 10Mois	20
الشهر 11Mois	20
الشهر 12Mois	20

المجموع =

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الثانية؟

بيع المنتج في السنة الثالثة Votre chiffre d'affaires de la troisième année

بيع المنتج في السنة الثالثة	متوسط أيام العمل في الشهر
الشهر 1Mois	20
الشهر 2Mois	20
الشهر 3Mois	20
الشهر 4Mois	20
الشهر 5Mois	20
الشهر 6Mois	20
الشهر 7Mois	20
الشهر 8Mois	20
الشهر 9Mois	20
الشهر 10Mois	20
الشهر 11Mois	20
الشهر 12Mois	20

المجموع =

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الثالثة ؟

تطور حجم رقم الأعمال في السنة

- النسبة المئوية للزيادة في حجم الأعمال بين السنة 1 و السنة 2؟
- النسبة المئوية للزيادة في حجم الأعمال بين السنة 2 و السنة 3 ؟

حاجتك لرأس المال العامل

30 يوم	متوسط مدة الاعتمادات الممنوحة للعملاء بالأيام Durée moyenne des crédits accordés aux clients en jours
365 يوم	متوسط مدة ديون الموردين بالأيام Durée moyenne des dettes fournisseurs en jours

رواتب الموظفين و مسؤولين الشركة

50.000 da / month	رواتب الموظفين Salaires employés
60.000 da / month	صافي أجور المسؤولين Rémunération nette dirigeant

Business Model Canvas

Designed for:	Designed by:	Date:	Version:

Key Partners
Who are our Key Partners? Who a
key suppliers? Which Key Resour
we acquiring from partners? Whicl
Activities do partners perform?

MOTIVATIONS FOR PARTNERS
Optimization and economy, Reduc
risk and uncertainty, Acquisition of
particular resources and activities

Key Activities
What Key Activities do our Value Propositions require? Our Distribu
Channels? Customer Relationship
Revenue streams?

CATEGORIES:
Production, Problem Solving, Platform/Network

Key Resources
What Key Resources do our Value Propositions require? Our Distribution Channels? Customer Relationships Revenue Streams?

TYPES OF RESOURCES: Physical, Intellectual (brand patents, copyrights, data), Human, Financial

Value Propositions
What value do we deliver to the cu
Which one of our customer's prob
we helping to solve? What bundle
products and services are we offe
each Customer Segment? Which needs are we satisfying?

CHARACTERISTICS: Newness, Performance, Customization, "Get
Job Done", Design, Brand/Status, Cost Reduction, Risk Reduction, Accessibility, Convenience/Usabil

Customer Relationships
What type of relationship does eac
Customer Segments expect us to
and maintain with them? Which or
we established? How are they inte
with the rest of our business mode
costly are they?

Channels
Through which Channels do our Customer Segments want to be reached? How are we reaching them now? How are our Channels integrated? Which ones work best? Which ones are most cost-efficient? How are we integrating them with customer routines?

Customer Segments
For whom are we creating value? Who are our most important customers? Is our customer base a Mass Market, Niche Market, Segmented, Diversified, Multi-sided Platform

Cost Structure
What are the most important costs inherent in our business model? Which Key Resour
most expensive? Which Key Activities are most expensive?

IS

BUSINESS MORE: Cost Driven (leanest cost structure, low price value proposition, ma
automation, extensive outsourcing), Value Driven (focused on value creation, premium
proposition).

SAMPLE

CHARACTERISTICS: Fixed Costs (salaries, rents, utilities), Variable costs, Economies
scale, Economies of scope

Revenue Streams
For what value are our customers really willing to pay? For what do they currently pay? How are they currently paying? How would they prefer to pay? How much does each Revenue Stream contribute to overall revenues?

TYPES: Asset sale, Usage fee, Subscription Fees, Lending/Renting/Leasing, Licensing, Brokerage fees, Advertising
FIXED PRICING: List Price, Product feature dependent, Customer segment dependent, Volume dependent
DYNAMIC PRICING: Negotiation (bargaining), Yield Management, Real-time-Market

Business Model Canvas

Designed for:	*Designed by:*	*Date:*	*Version:*
	Boulefred Yassmina Boumaza Chaimaa	-28/05/2024	

Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments
• Universities • Academic institutions	• Continuous improvement of NLP algorithms • Maintenance and updates for the chatbot **Key Resources** • NLP technology • Student databases • Developers and AI specialists • Server infrastructure for hosting	• Personalized student assistance • Context-aware information delivery • 24/7 availability • Integration with existing academic systems	• Automated support through the chatbot • Feedback mechanisms for continuous improvement **Channels** • Web application • Mobile application	• Students • Academic institutions • Faculty and staff

Cost Structure

- Development and maintenance costs
- Server and hosting expenses
- Marketing and promotional costs
- Employee salaries and training

Revenue Streams

- Subscription model for academic institutions
- Freemium model for students with premium features
- Licensing the chatbot technology to other institutions

Bibliography

[1] E. Adamopoulou et L. Moussiades. « An overview of chatbot technology ». In : *IFIP international conference on artificial intelligence applications and innovations*. Springer. 2020, p. 373-383 (cf. p. 41).

[2] E. W. Aditya, S. Ismail et N. Allias. « Implementation of Intelligent Chatbot in Student Portal : A Systematic Literature Review ». In : *2022 International Visualization, Informatics and Technology Conference (IVIT)*. IEEE. 2022, p. 47-51 (cf. p. 42).

[3] A. Aloqayli et H. Abdelhafez. « Intelligent Chatbot for Admission in Higher Education ». In : *International Journal of Information and Education Technology* 13.9 (2023), p. 1348-1357 (cf. p. 40).

[4] A. Argal et al. « Intelligent travel chatbot for predictive recommendation in echo platform ». In : *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*. IEEE. 2018, p. 176-183 (cf. p. 42).

[5] J.-B. Aujogue et A. Aussem. « Hierarchical Recurrent Attention Networks for Context-Aware Education Chatbots ». In : *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, p. 1-8 (cf. p. 6, 42).

[6] Bird, Steven, Klein, Ewan et Loper, Edward. *Natural Language Toolkit (NLTK)*. Accessed on [date you accessed the website]. 2001. url : `https://www.nltk.org/` (cf. p. 65).

[7] C. K. Y. Chan et W. Hu. « Students' Voices on Generative AI : Perceptions, Benefits, and Challenges in Higher Education ». In : *arXiv preprint arXiv :2305.00290* (2023) (cf. p. 42).

[8] Y. Chen et al. « Artificial intelligence (AI) student assistants in the classroom : Designing chatbots to support student success ». In : *Information Systems Frontiers* 25.1 (2023), p. 161-182 (cf. p. 42).

[9] F. Clarizia et Others. « A context-aware chatbot for tourist destinations ». In : *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2019, p. 348-354 (cf. p. 42).

[10] M. Crump. *Sketching your app design*. Accessed : 2024-06-05. 2016. url : `https://blogs.windows.com/windowsdeveloper/2016/05/17/sketching-your-app-design/` (cf. p. 75).

[11] e. Eleonora Pura. « Designing a Chatbot ». Thèse de doct. University of Zurich, 2021 (cf. p. 37-39, 41).

[12] K. D. Foote. « A brief history of natural language processing (nlp) ». In : *DATAVERSITY, May* 22 (2019) (cf. p. 14).

[13] P. S. Foundation. *json — JSON encoder and decoder*. Python 3.12.3 Documentation. url : `https://docs.python.org/3/library/json.html` (cf. p. 65).

[14] P. S. Foundation. *Python Documentation*. Accessed on [date you accessed the documentation]. url : https://docs.python.org/ (cf. p. 65).

[15] P. Foy. « Understanding Transformers the Architecture of LLMs ». In : *mlq.ai* (2024) (cf. p. 24).

[16] Y. Gao et al. « Retrieval-augmented generation for large language models : A survey ». In : *arXiv preprint arXiv :2312.10997* (2023) (cf. p. 27-29).

[17] GeeksforGeeks. *Libraries in Python*. Accessed : 2024-06-05. 2023. url : https://www.geeksforgeeks.org/libraries-in-python/ (cf. p. 65).

[18] A. Gupta et al. « Casa-nlu : Context-aware self-attentive natural language understanding for task-oriented chatbots ». In : *arXiv preprint arXiv :1909.08705* (2019) (cf. p. 42).

[19] M. U. Hadi et al. « Large language models : a comprehensive survey of its applications, challenges, limitations, and future prospects ». In : *Authorea Preprints* (2023) (cf. p. 29).

[20] S. Han et M. K. Lee. « FAQ chatbot and inclusive learning in massive open online courses ». In : *Computers & Education* 179 (2022), p. 104395 (cf. p. 42).

[21] H. Hassan et al. « Review and classification of AI-enabled COVID-19 CT imaging models based on computer vision tasks ». In : *Computers in biology and medicine* 141 (2022), p. 105123 (cf. p. 66).

[22] Y. Ikemoto et Others. « Conversation strategy of a chatbot for interactive recommendations ». In : *Intelligent Information and Database Systems : 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part I 10*. Springer. 2018, p. 117-126 (cf. p. 42).

[23] N. Kandpal et al. « Large language models struggle to learn long-tail knowledge ». In : *International Conference on Machine Learning*. PMLR. 2023, p. 15696-15707 (cf. p. 26).

[24] S. Khandare. *Mastering Large Language Models : Advanced techniques, applications, cutting-edge methods, and top LLMs (English Edition)*. Bpb Publications, 2024. isbn : 9789355519658. url : https://books.google.dz/books?id=xp_6EAAAQBAJ (cf. p. 20).

[25] V. Khobragade. « The Journey of Large Language Models : Evolution, Application, and Limitations ». In : *medium* (2024) (cf. p. 26, 30).

[26] D. Khurana et al. « Natural language processing : State of the art, current trends and challenges ». In : *Multimedia tools and applications* 82.3 (2023), p. 3713-3744 (cf. p. 22).

[27] M. A. Kuhail et al. « Interacting with educational chatbots : A systematic review ». In : *Education and Information Technologies* 28.1 (2023), p. 973-1018 (cf. p. 42).

[28] E. Kumar. *Natural Language Processing*. I.K. International Publishing House Pvt. Limited, 2013. isbn : 9789380578774. url : https://books.google.dz/books?id=FpUBFNFuKWgC (cf. p. 12).

[29] E. D. Liddy. « Natural language processing ». In : (2001) (cf. p. 15-17).

[30] Looka. *15 Wireframe Examples and How to Make Your Own*. Accessed : 2024-06-05. 2023. url : https://looka.com/blog/wireframe-examples/ (cf. p. 76).

[31] S. K. Nagarajan. « Server-Less Rule-Based Chatbot Using Deep Neural Network ». Thèse de doct. UPPSALA UNIVERSITET, 2019 (cf. p. 34).

[32] I. Nica, O. A. Tazl et F. Wotawa. « Chatbot-based Tourist Recommendations Using Model-based Reasoning. » In : *Conf WS*. 2018, p. 25-30 (cf. p. 42).

[33] C. W. Okonkwo et A. Ade-Ibijola. « Chatbots applications in education : A systematic review ». In : *Computers and Education : Artificial Intelligence* 2 (2021), p. 100033 (cf. p. 42).

[34] A. Ouared, M. Amrani et P. Schobbens. « A Context-Aware Chatbot for Student Assistance Services in Higher Education ». In : *Proceedings of the 16th International Conference on Computer Supported Education, CSEDU 2024, Angers, France, May 2-4, 2024, Volume 1.* Sous la dir. d'O. Poqet et al. SCITEPRESS, 2024, p. 264-271 (cf. p. 56).

[35] A. Ouared, M. Amrani et P. -Y. Schobbens. « Learning Analytics Solution for Monitoring and Analyzing the Students' Behavior in SQL Lab Work ». In : (2023) (cf. p. 6).

[36] A. Ouared et A. Chadli. « Using MDE for Teaching Database Query Optimizer. » In : *ENASE*. 2021, p. 529-536 (cf. p. 6).

[37] S. N. M. S. Pi et M. A. Majid. « Components of Smart Chatbot Academic Model for a University Website ». In : *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*. IEEE. 2020, p. 1-6 (cf. p. 42).

[38] R. Python. *Working With JSON Data in Python*. url : `https://realpython.com/courses/working-json-data-python/` (cf. p. 65).

[39] M. A. K. Raiaan et al. « A review on large Language Models : Architectures, applications, taxonomies, open issues and challenges ». In : *IEEE Access* (2024) (cf. p. 25).

[40] K. Sevnarayan. « The Implementation of Telegram as A Pedagogical Tool to Enhance Student Motivation and Interaction ». In : *Journal of Education Technology* 7.1 (2023) (cf. p. 42).

[41] A. Shevat. *Designing bots : Creating conversational experiences*. " O'Reilly Media, Inc.", 2017 (cf. p. 6).

[42] J. Singh, M. H. Joesph et K. B. A. Jabbar. « Rule-based chabot for student enquiries ». In : *Journal of Physics : Conference Series*. T. 1228. 1. IOP Publishing. 2019, p. 012060 (cf. p. 42).

[43] T. Sreemany. « Essential Text Pre-processing Techniques for NLP ! » In : *analytics vidhya* (2022) (cf. p. 19).

[44] O. Talbi et A. Ouared. « Goal-oriented student motivation in learning analytics : How can a requirements-driven approach help ? » In : *Education and Information Technologies* 27.9 (2022), p. 12083-12121 (cf. p. 6).

[45] S. Tegos et S. Demetriadis. « Conversational agents improve peer learning through building on prior knowledge ». In : *Journal of Educational Technology & Society* 20.1 (2017), p. 99-111 (cf. p. 42).

[46] D. Trifunovic. « NLP-based chatbot for HAMK ». In : (2019) (cf. p. 34).

[47] A. Vaswani et al. « Attention is all you need ». In : *Advances in neural information processing systems* 30 (2017) (cf. p. 24).

[48] K. Venusamy et R. K. Basha. « Design and Implementation of FAQ Chabot in Higher Education Institutions web portal-Sultanate of Oman ». In : *Int. J. of Research and Analytical Reviews (IJRAR)* 8.4 (2021), p. 858-861 (cf. p. 42).

[49] R. Winkler et al. « Sara, the lecturer : Improving learning in online education with a scaffolding-based conversational agent ». In : *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, p. 1-14 (cf. p. 42).

[50] R. Meyer von Wolff et al. « Chatbots for the information acquisition at universities–a student's view on the application area ». In : *Chatbot Research and Design : Third International Workshop, CONVERSATIONS 2019, Amsterdam, The Netherlands, November 19–20, 2019, Revised Selected Papers 3*. Springer. 2020, p. 231-244 (cf. p. 42).

[51] Y. Zhang et al. « Siren's song in the AI ocean : a survey on hallucination in large language models ». In : *arXiv preprint arXiv :2309.01219* (2023) (cf. p. 27).