People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Tahri Mohamed – Bechar
Faculty of Exact Sciences
Department of Mathematics and Computer Science

Final Year Project

In partial fulfillment of the requirements for the Bachelor's degree in Computer Science

Option: Information Systems

Topic

Development of a Web Application for Textual File Processing with AI

Presented by: Gasba Ahmed
Ouahabi Abdessamed

Supervised by: Dr. Khair Younes

Contact: ahmedgasba4@gmail.com | ouahabi.abdessamed@univ-bechar.dz

Academic year: 2024/2025

ACKNOWLEDGEMENTS

First of all, we want to thank Allah, for giving us the strength, patience, and determination to finish this work.

We would like to sincerely thank our supervisor, Dr. Khair Younes, for his kind agreement to guide our final year project. His valuable advice, unwavering support, and insightful guidance were instrumental in the successful completion of this work.

Our heartfelt thanks also go to the jury members for their time, attention, and willingness to evaluate our research and efforts. We are equally grateful to all the professors of Tahri Mohammed University of Bechar, particularly those in the MI Department, as well as the dedicated staff who contributed to creating a supportive learning environment.

Finally, we want to say a big thank you to our parents and grandparents for always taking care of us, supporting us, and praying for us. Their steady support has helped and inspired us during our academic journey.

Dedication

Firstly, thanks to Allah for everything. Then i would like to express my sincere thanks to our supervisor Dr. Khair Younes for his guidance, support, and valuable advice throughout this journey.

My deepest gratitude goes to my beloved family my mother, my father Gasba Abdenacer, my sister and my brother Gasba Mohammed Badereddine for their unwavering support, motivation, and constant belief in me.

I extend my heartfelt appreciation to my friends Ouahabi Abdessamed, khader Faycal, Othmane Houssine, Hachimi Mohamed, Guellil Abd Eljalil for their encouragement, constructive feedback, and companionship during this project.

Lastly, I thank all those who contributed in any way to the success of this project, and I sincerely acknowledge the jury members for taking the time to evaluate my work.

Ahmed.

Dedication

First and foremost, all praise is due to Allah for His guidance and blessings, without which this achievement would not have been possible.

I would like to express my sincere gratitude to all the professors who accompanied us throughout our academic journey for their knowledge, support, and guidance. I especially thank Dr. Khair Younes for his valuable advice, encouragement, and continuous support, which had a significant impact on the success of this project.

I also extend my heartfelt appreciation to my beloved family especially my dear mother, my caring sister, and my brother Abdelmadjid for their unwavering support, motivation, and belief in me. I offer a special prayer and remembrance for the soul of my late father, Abdelkader, who will always remain a source of inspiration and pride.

Lastly, I would like to thank all my friends for their encouragement, companionship, and support throughout this journey.

With sincere appreciation,

Abdessamed.

Abstract:

In recent years, PDF files have become more important in our lives, and sometimes reading these files is time-consuming and exhausting, especially when it is an eBook with more than page and maybe contains outdated information, or when users are looking for specific answers or summaries. Traditional keyword search tools are limited in understanding context and can overwhelm users with irrelevant results. To address this problem, we developed a web-based AI chatbot that allows users to interact with PDF documents through natural language. By combining LangChain, Streamlit, and powerful Large Language Models (LLMs) such as GPT-3.5 Turbo and Llama 3, and Natural Language Processing (NLP) techniques, our system can understand user questions, extract relevant content, summarize sections, and provide accurate, conversational responses. This solution simplifies document navigation, improves accessibility, and demonstrates the potential of conversational AI in intelligent document processing.

Keywords: Chatbot**,** Large Language Model (LLM), Natural Language Processing (NLP), Artificial Intelligence (AI), PDF.

Résumé:

Ces dernières années, les fichiers PDF sont devenus essentiels dans notre vie quotidienne. Cependant, la lecture de ces fichiers peut être longue et fatigante, surtout lorsqu'il s'agit d'un eBook de plus d'une page, contenant parfois des informations obsolètes, ou lorsque l'utilisateur recherche des réponses précises ou des résumés. Les outils de recherche traditionnels par mots-clés sont limités dans leur compréhension du contexte et peuvent submerger l'utilisateur avec des résultats non pertinents. Pour répondre à ce problème, nous avons développé un chatbot basé sur l'IA accessible via le web, qui permet aux utilisateurs d'interagir avec des fichiers PDF en langage naturel. En combinant LangChain, Streamlit, et des Modèles de Langage de Grande Taille tels que GPT-3.5 Turbo et Llama 3, et les techniques du Traitement Automatique du Langage Naturel (TALN), notre système est capable de comprendre les questions des utilisateurs, d'extraire les informations pertinentes, de résumer le contenu et de fournir des réponses précises et conversationnelles. Cette solution facilite la navigation dans les documents, améliore l'accessibilité à l'information et démontre le potentiel de l'IA conversationnelle dans le traitement intelligent de documents.

Mot clé: Chatbot**,** Modèle de Langage de Grande Taille (LLM), Traitement Automatique du Langage Naturel (TALN), Intelligence Artificielle (IA), PDF

ملخص:

أصبحت الملفات الالكترونية PDF في السنوات الأخيرة أكثر أهمية في حياتنا اليومية. ومع ذلك، فإن قراءة هذه الملفات قد تستغرق وقتًا طويلاً وتُسبب الإرهاق، خصوصًا عند التعامل مع كتب إلكترونية تحتوي على أكثر من صفحة وقد تتضمن معلومات قديمة، أو عندما يبحث المستخدم عن إجابات محددة أو ملخصات، أدوات البحث التقليدية المعتمدة على البحث بالكلمات المفتاحية تعاني من محدودية في فهم السياق، مما يؤدي إلى نتائج غير ملائمة. لمواجهة هذا التحدي، قمنا بتطوير روبوت دردشة ذكي يعمل على الويب، يتيح للمستخدمين التفاعل مع ملفات PDF، تم بناء هذا النظام باستخدام LangChain و Streamlit، بالإضافة إلى نماذج لغوية ضخمة مثل GPT-3.5 Turbo و Llama 3، و أدوات لمعالجة اللغة الطبيعية التي مكنت هذا النظام من فهم الأسئلة، واستخراج المعلومات، وتلخيص محتوى المستندات، و تقديم إجابات دقيقة. سهل هذه

النظام التنقل في المستندات، وتحسن الوصول إلى المعلومات، وبرز إمكانيات الذكاء الاصطناعي التفاعلي في معالجة المستندات بذكاء.

كلمات مفتاحية: روبوت دردشة، نماذج لغوية ضخمة، معالجة اللغة الطبيعية، الذكاء الاصطناعي، ملفات الكترونية PDF

List of abbreviations:

PDF: Portable Document Format.

AI: Artificial Intelligence.

LLM: Large Language Model.

NLP: Natural Language Processing.

ML: Machine Learning.

DL: Deep Learning.

ANN: Artificial Neural Network.

POS: Part Of Speech.

NLU: Natural Language Understanding.

NLG: Natural Language Generation.

NER: Named Entity Recognition.

FAISS: Facebook AI Similarity Search

Table of Contents

List of Figures

General introduction

General Introduction:

In the digital age, a vast amount of data is stored in electronic documents, PDFs are the most commonly used format across different domains. Researchers, educators, students, businesses, and legal professionals rely on PDF files to store reports, academic papers, manuals, contracts, etc. However, extracting specific data from these files remains a complex challenge. Many PDFs contain unstructured or semi-structured text, making manual reading and comprehending content challenging and time-consuming. This issue becomes even more problematic when dealing with multiple PDFs or large documents requiring quick access to specific information.

Artificial Intelligence has improved numerous fields, including document processing. Through advancements in Machine Learning (ML), Natural Language Processing (NLP), Deep Learning (DL), and Large Language Models (LLMs), AI enables efficient text analysis, extraction, and understanding. AI-based solutions can automate multiple tasks such as question answering, summarization, and entity recognition, greatly improving productivity. These innovations have led to intelligent systems capable of accurately processing and understanding human language [2].

The difficulty in retrieving, summarizing, and understanding PDF content highlights a critical problem: How can users efficiently interact with large-volume PDFs without manual intervention? Traditional search functions in PDFs often fall to answering complex questions or providing summaries. Users may struggle navigating through long documents, identifying key insights, or extracting relevant information [2].

To address this problem, we propose the development of an **AI-powered web-based chatbot** that processes PDF documents. This chatbot enables users to:

- Upload one or multiple PDFs for automated processing.
- Ask questions related to the PDFs and receive AI-generated answers.
- Extract specific data from the documents, such as dates, names, statistics, or other details.
- Summarize entire documents or specific parts to quickly grasp the main ideas.
- Explain complex parts of the text in simpler terms for better understanding.

This system utilizes NLP models and Large Language Models (LLMs) to interpret queries, retrieve relevant sections from documents, and generate accurate responses. By integrating AI-based text summarization, extraction, and question-answering capabilities, our chatbot offers users an easy and efficient way to interact with PDFs without manually searching through large documents [2].

The primary objective of our project is to develop a user-friendly web application that facilitates PDF processing. Specifically, our project aims to:

1. Automate data retrieval from PDFs using AI-driven techniques.
2. Enhance accessibility by enabling users to query documents naturally rather than keyword searches.
3. Improve efficiency by extracting essential information and summarizing large PDFs.
4. Ensure accuracy in responses by using NLP models and LLMs.

To provide a detailed overview of our research and development process, this document is structured as follows:

- **Chapter 1:** Foundation of AI, its applications in different domains, focusing on NLP and LLMs in text processing.
- **Chapter 2:** Covers the foundation of chatbots, and why NLP is essential in chatbots, and how to build one, without forgetting the challenges in chatbots, and finalizes it with some existing research.
- **Chapter 3:** Presents the chatbot architecture design and the implementation details, focusing on the technologies used and how the system was built, and shows the chatbot PDF results.
- **General conclusion:** Summarizes our research and offers recommendations for future project improvement.

In this research and implementation, we aim to demonstrate the power of AI in textual file processing and how efficiently it can handle PDF files. This project serves as an innovative step toward intelligent document analysis, improving accessibility in various fields such as research, education, and business.

Chapter I: Introduction to Artificial Intelligence

Chapter I: Introduction to Artificial Intelligence

1. Introduction:

The infusion of artificial intelligence in modern technology is inevitable. It changes how humans interact with machines. So far, AI has proven to increase productivity in several areas through automating repetitive work or establishing complex decision-making processes. Among these abilities, the ability to process and analyze text data is one of the most critical. Texts, these days, have become some of the most influential pieces of information, especially in a time when people are starting to part from structured formats and embrace more forms like semi-structured to unstructured ones, such as PDF. They become difficult to extract, comprehend, and summarize as input in most applications. This forms one of the prime warrants of importance for AI in the present times.[1]

This chapter provides an overview of Artificial Intelligence, its evolution, and its application, focusing on text extraction and document processing. It begins with an AI definition and explores its key components, such as Deep Learning (DL), Machine Learning (ML), and Natural Language Processing (NLP). These important technologies enable tasks like text understanding, summarization, and question answering. The chapter also highlights the role of AI in various domains, like healthcare, finance, and education, noting its versatility.[1][2]

The main theme in this chapter is the application of AI in text extraction, a process that involves extracting valuable information from documents. Traditional methods, such as OCR and rule-based systems, have limitations in handling unstructured text. In contrast, AI-based methods utilize advanced models like LLAMA and GPT to achieve higher accuracy. These models, built on transformer architectures, have changed NLP by enabling machines to understand context, extract entities, and generate responses similar to human conversation.[2][3]

Despite its advancement, AI faces text extraction challenges, including accuracy, costs, and ethical concerns such as data privacy. By noticing these challenges, we can make it faster, more accurate, and accessible. This chapter will discuss the technical aspects of AI-driven text extraction.[1][3]

2. Artificial Intelligence:

2.1. Definition:

Artificial Intelligence (AI) is a branch of computer science that aims to make computers perform tasks that usually require human thinking. These tasks involve using your brain to think, learn, solve problems, look at things, and understand words. AI combines ideas from math, statistics, thinking, brain science, and language to create machines and software that can think and behave like people. The phrase "Artificial Intelligence" was first used by John McCarthy in 1956 during a meeting known as the Dartmouth Conference. This meeting is considered the beginning of AI as a field of study. Since then, AI has changed from basic systems that follow instructions to very advanced models that can learn from information and get better on their own. AI systems are made to act like human thinking. This means machines can learn from what they experience, adapt to new information, and perform tasks on their own, like people do. This technique of copying how people think has made AI very useful in areas like healthcare, finance, education, robotics, language understanding, and text processing. There are two main methods to make AI systems: symbolic AI, which uses rules to follow, and sub symbolic AI, which doesn't follow rules in the same way. Symbolic AI relies on clear rules and logic, while sub symbolic AI, which is used more often today, learns from data using models like neural networks. It picks up patterns and makes changes without needing exact directions. One key thing about AI is that it can learn and improve over time. This skill, known as machine learning, allows AI systems to learn from their past experiences and get better at tasks on their own without needing ongoing help from people. Better systems can think more carefully and correct their own errors, increasing what machines are capable of. AI is usually divided into three types based on what it can do.

- Narrow AI.
- General AI.
- Super AI.

In our project, AI helps us make smart systems that can read, understand, and answer questions about text in PDFs using simple language. We will explore this topic further in the next chapters [6][7][9].

3. A Brief History of AI:

The beginnings of artificial intelligence (AI) come from important ideas in math and computer science. Many researchers believe that the basic ideas for machines that can think like humans began with Leibniz, who wanted to turn human thinking into calculable steps, and George Boole, whose logic system is the foundation of today's computing [11].

In the area of AI, important early achievements include the work of mathematicians like Alan Turing and Walter Pitts. In 1943, Pitts worked with brain doctor Warren McCulloch to create the first computer model based on how the human brain's networks work. They combined Turing's binary logic, Bertrand Russell's propositional logic and Sherrington's studies on how synapses work. they designed networks of simple logic gates (AND, OR, NOT) that could be interconnected to represent complex propositions [9][10].

In 1950, Turing continued the conversation about computers with his important paper, "Computing Machinery and Intelligence." In it, he introduced the "imitation game," which later became known as the Turing Test. This test checks if a machine can act smart in a way that is hard to tell apart from how a human would act. The Turing Test focus on observable behavior, suggesting that if a machine can convincingly imitate a human in conversation, it could be considered intelligent. However, it is still an important part of the study of artificial intelligence [9][11].

In the 1950s, two areas of research came together: one was about how behavior adapts, known as cybernetics introduced by Norbert Wiener in 1948, and the other was about using symbols for calculations, called symbolic AI. These split into different areas, with cybernetics focusing on living systems and symbolic AI manipulating symbols to represent knowledge and reasoning [9][10][11].

In 1969, Marvin Minsky and Seymour Papert published Perceptrons, a critical analysis of early neural networks. Their work demonstrated that single-layer perceptrons were mathematically limited and incapable of solving certain basic tasks, such as recognizing when two inputs differ. Although their analysis did not apply to multilayer networks. Ironically, algorithms like backpropagation, which would later revive neural network research in the 1980s, had already been proposed as early as 1969 [9].

The comeback of AI in the late 1990s and early 2000s, driven by the growth of the internet and better computer power, was an important change. The arrival of big data made it easier to train AI systems more accurately. Events like IBM's Deep Blue beating chess champion Garry Kasparov showed what AI can do. Projects like ImageNet and OpenAI helped the field grow by offering important data and computer programs [8][11].

Today, AI algorithms are everywhere, used in areas like image processing, processing textual data, understanding language, recognizing voices, and making predictions [8].

4. Types of artificial intelligence:

4.1. Types of artificial intelligence based on capabilities:

AI as a transformative technology, can be categorized based on its capabilities to perform intelligent tasks. This classification provides a framework for understanding the different operational domains of AI, which is essential in various applications including processing PDF files. The three primary classifications of AI include Narrow AI, General AI, and Super AI:

- **Weak or Artificial Narrow Intelligence (ANI):** A weak or narrow AI is a type of AI that performs assigned tasks using intelligence. It is the most common form of AI available in industries today. Narrow AI cannot operate beyond what is assigned to the system. Indeed, it is trained to perform a single specific task. ANI stands for all artificial intelligence machines created and deployed to date. All artificially intelligent systems that can perform a dedicated task autonomously using human-like capabilities fall into this category. As the name suggests, these machines have very limited responsibilities.

  The assertion that machines could act as if they were intelligent weak AI is called the weak AI or narrow AI hypothesis. Narrow artificial intelligence (narrow AI) is a specific type of artificial intelligence in which a learning algorithm is designed to perform a single task, and any knowledge gained from performing that task will not automatically be applied to other tasks. Most AI applications in use today can be categorized as being narrow AI [8].
- **Artificial General Intelligence (AGI):** Artificial general intelligence is a type of AI capable of performing any intellectual task as a human being. AGI machines are designed to perceive, learn and function entirely like humans. AGI systems are machines capable of reproducing human multifunction capabilities. AGI are designed to create multiple skills that can significantly reduce the time needed to train these

machines. Many philosophers have claimed that a machine that passes the Turing Test would still not be actually thinking, but would be only a simulation of thinking. The assertion that machines that could act as if they were intelligent called STRONG AI are actually thinking (not just simulating thinking) is called the strong AI hypothesis. The concept of strong artificial intelligence refers to a machine capable not only of producing intelligent behavior, in particular of modeling abstract ideas, but also of experiencing an impression of real consciousness, of "real feelings" (although one can put behind these words), and understanding of ones own reasonings [8][9].

- **Super artificial intelligence (ASI):** Artificial super-intelligent systems can be described as the zenith of AI research. ASI is intended not only to reproduce multi-faceted human intelligence, but also to have faster memory, data processing, and analysis capabilities. It is a hypothetical concept of artificial intelligence in which researchers try to develop machines capable of surpassing humans. This is the result of General AI [8][9].

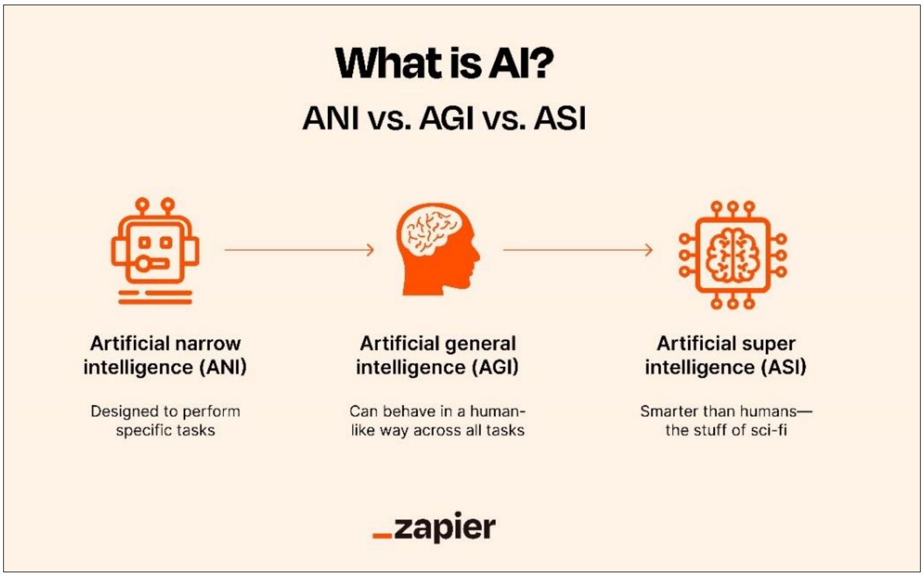

What is AI?
ANI vs. AGI vs. ASI
Artificial narrow intelligence (ANI)
Designed to perform specific tasks
Artificial general intelligence (AGI)
Can behave in a human-like way across all tasks
Artificial super intelligence (ASI)
Smarter than humans—the stuff of sci-fi
_zapier

Figure I.1 AI Evolution: Understanding ANI, AGI, and ASI [40]

4.2. Types of artificial intelligence based on functionality:

- **Reactive machines:** These are the simplest AI systems. They don’t remember things or use old experiences to affect their choices later. They give set answers to certain questions or actions. One example is IBM's Deep Blue, which could play chess but couldn't remember things or learn new skills [8].
- **Limited memory:** Limited-memory machines are those that can retain memory for a short period of time. These machines have the same capabilities as those of purely reactive machines. Moreover, machines with limited memory can learn from previous experiences to make decisions. For example, self-driving cars are limited-memory machines that can store data such as the distance the car has traveled with nearby cars, their recent speed, speed limit, and lane [8].
- **Theory of mind:** Theory of mind can be defined as a simulation. Let’s be clear, when a person thinks of themselves as another person, their brain tends to simulate

that other person’s mind. Theory of mind is essential to human cognition. Moreover, it is also crucial for social interactions. A break in the concept of theory of mind, for example, can be illustrated as a case of autism [8].

- **Self-aware AI:** Self-awareness machines hypothetically exist today. As the name suggests, these machines are supposed to be aware of themselves, like the human brain. Machines can be described as the ultimate goal of AI scientists. The goal of developing self-awareness machines is to make them capable of generating emotions and needs identical to those of humans [8].

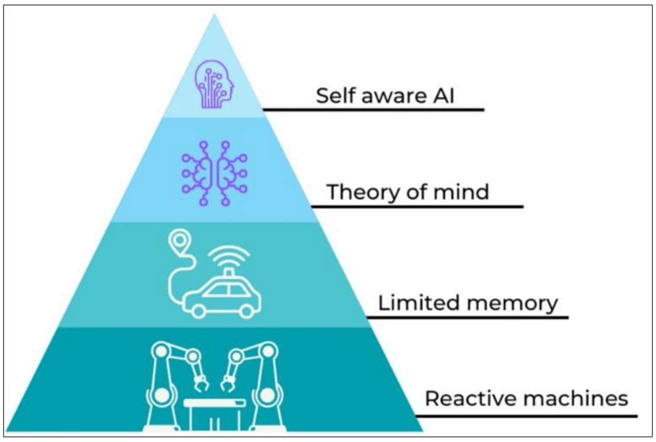


Figure I.2 The AI Hierarchy: From Reactive Machines to Self-Aware Systems [41]

5. Subsets of artificial intelligence:

5.1. Machine Learning (ML):

Machine Learning (ML) is an important part of Artificial Intelligence. It helps computers learn from data and get better at tasks over time without needing specific instructions. It shows a move from old rule-based systems to new methods that use data, allowing the system to change and improve based on patterns seen in large amounts of information.

At its heart, machine learning is about finding knowledge on its own. An ML model is taught using past data, so it understands the connections between what goes in and what comes out. After it learns, it can use that knowledge to make guesses or choices when it encounters new information that it hasn't seen before. This ability to generalize is what makes machine learning different from fixed systems [8].

Types of Machine Learning used:

- **Supervised learning:** means teaching a model using a set of data that has both the questions and the correct answers. The system learns to connect inputs to outputs, which makes it great for tasks like sorting things (for example, finding spam emails) and making predictions (like estimating house prices) [1][8].

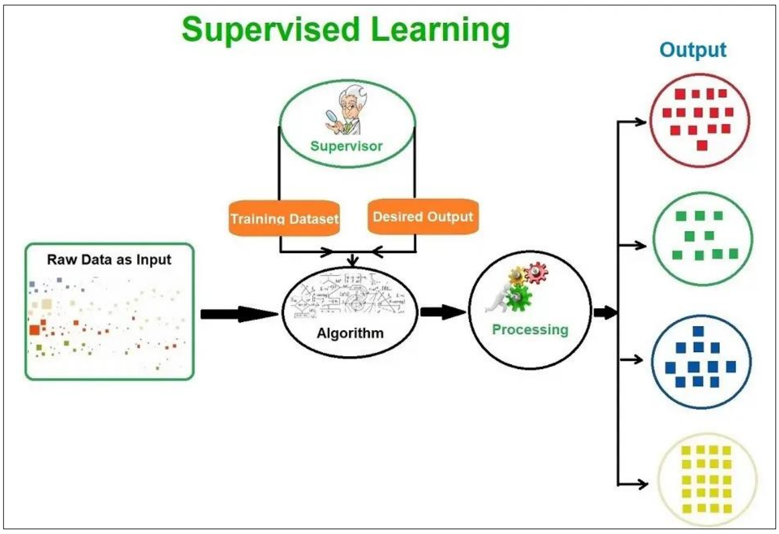


Figure I.3 Supervised Machine Learning [42]

- **Unsupervised learning:** works with data that doesn't have labels. The aim is to find hidden patterns or groups in the data, like how customers behave similarly or to simplify the data using methods like Principal Component Analysis (PCA) [8].

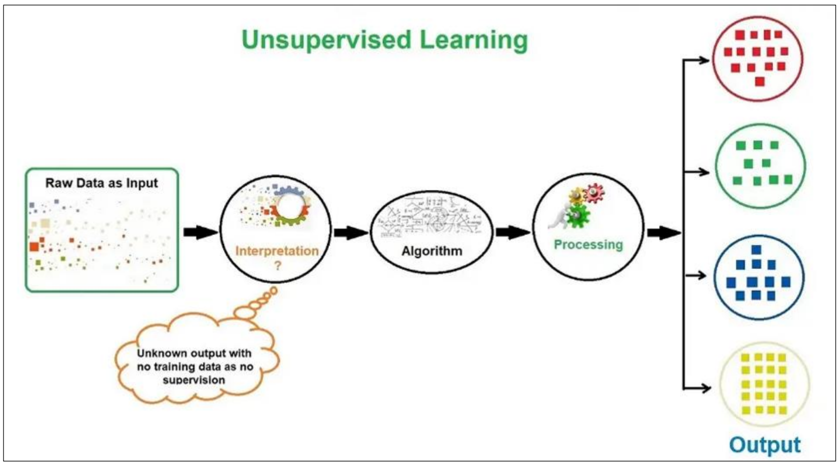


Figure I.4 Unsupervised Machine learning [42]

- **Reinforcement learning:** is when a computer program learns by trying things out in a setting and getting rewards or punishments for its actions. It is especially helpful in situations where quick decisions are needed, such as in robots and video game AI [1][8].

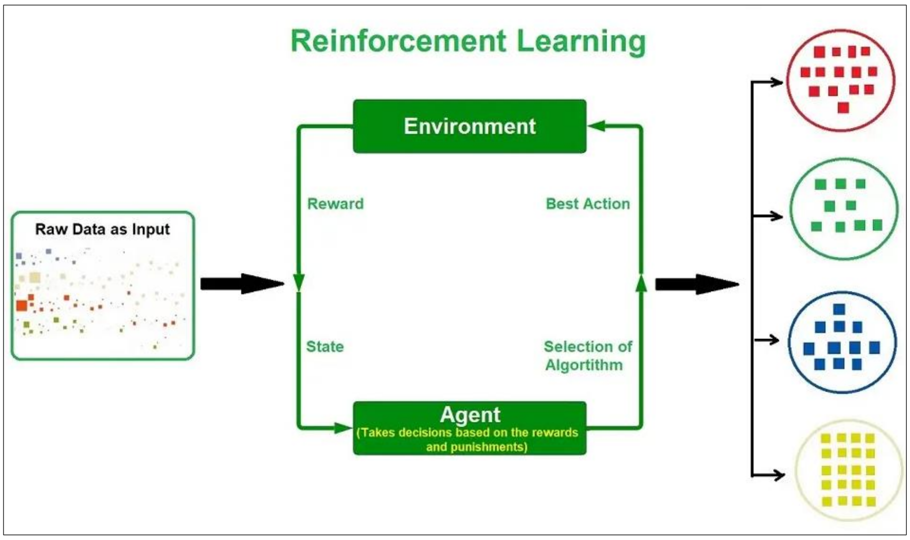


Figure I.5 Reinforcement learning [42]

Machine learning is commonly used in everyday tasks like suggesting products, finding fraud, understanding spoken words, and processing language. Its success usually depends on the quality of the data, the selection of features, and the computer power available. Machine learning is an important part of today's AI systems. It helps machines act smart[8].

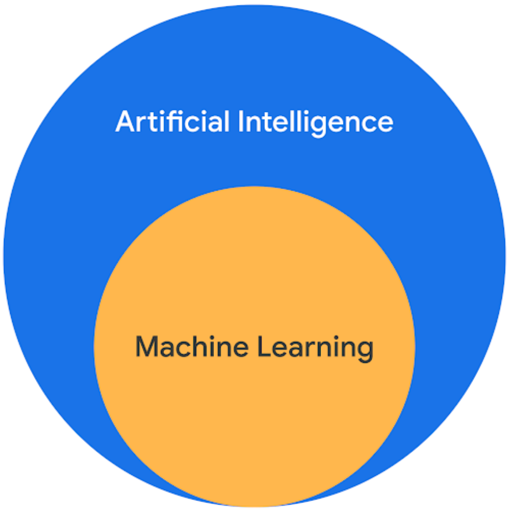


Figure I.6 Relationship between Artificial intelligence and machine learning [43]

5.2. Artificial Neural Network (ANN):

Artificial Neural Networks (ANNs) are a type of computer model used in machine learning that is based on how the human brain works. An ANN is made up of connected parts called neurons. These parts work together in layers to understand and learn complex patterns in data. These models are the basis for many advanced AI systems and are especially good at solving problems that are hard to explain with regular methods [4][8].

An artificial neural network typically consists of three types of layers:

- **Input layer:** which gets the starting data [4].
- **Hidden layers**: change information using weighted connections and activation functions [4].
- **Output layer**: which gives the final answer or prediction [4].

Every neuron in one layer connects to neurons in the next layer, and each connection has a weight that changes as the model learns. The network adjusts these weights using a method called backpropagation to reduce mistakes in its predictions. This learning process helps the ANN get better as it goes along [4].

Neural networks are very good at dealing with complex and large sets of data. They are often used in areas like recognizing pictures, understanding language, and studying biological data. For example, an ANN can learn to identify handwritten numbers or change text from one language to another [4][8].

One of the main benefits of ANNs is that they can easily adjust and grow. They can closely match nearly any function if they have enough data and time to learn. But there are also some problems, like needing a lot of data, taking a long time to train, and it's hard to understand how they work inside, which is often called the "black box" problem [4][8].

In summary, ANNs are strong tool in AI, can learn and represent complex patterns, and are crucial in developing intelligent systems [4][8].

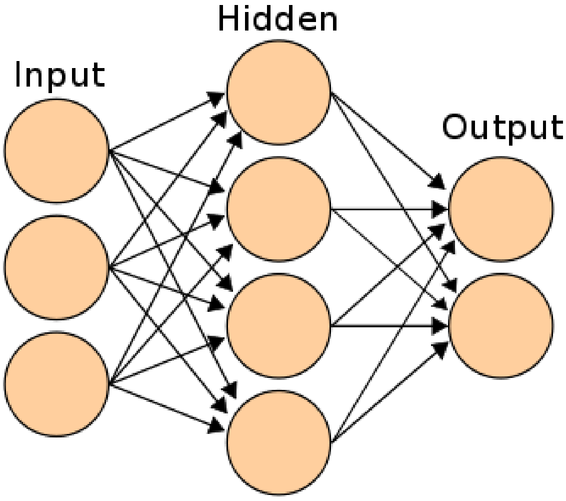


Figure I.7 Basic architecture of an ANN [44]

5.3. Deep Learning (DL):

Deep Learning is a more advanced part of machine learning that uses complicated networks with many layers to understand complex patterns in data. It is based on how people learn in steps, starting with simple ideas and building up to more complex ones [1][8].

What separates deep learning from traditional machine learning is its depth-these models comprise of various interconnected layers that permit them to memorize highlights naturally without the requirement for manual include building. Deep learning models are competent of taking care of tremendous sums of unstructured information such as pictures, sound, and content, making them vital in today's AI scene [1][8].

Some common types of deep learning structures are:

- **Convolutional Neural Networks (CNNs):** are commonly used for tasks like processing images, finding objects, and understanding pictures with computers [5].
- **Recurrent Neural Networks (RNNs):** are great for working with data that comes in a sequence, like talking, writing, and predicting future numbers based on past patterns [5].
- **Transformers:** are crucial for today's natural language processing, which includes models like BERT and GPT [3][7].

Deep learning has led to important progress in many areas. In healthcare, it helps look at medical images and find out if someone is sick. In self-driving cars, advanced computer programs analyze pictures from cameras to make quick driving choices. In the area of understanding language, deep learning helps with virtual assistants, translating languages, and summarizing documents [1][5][8].

Even though deep learning is powerful, it has some limits. These models need a lot of computer power and a big amount of labeled data. They can also fit too closely to the training data if not managed properly, and it's often hard to understand how they make decisions. This can be a problem in important areas like healthcare or finance [1][8].

Nevertheless, deep learning represents one of the most powerful tools in artificial intelligence, pushing the boundaries of what machines can learn and perform. As computational resources continue to grow, deep learning is expected to remain at the forefront of AI research and application [1][5] [8].

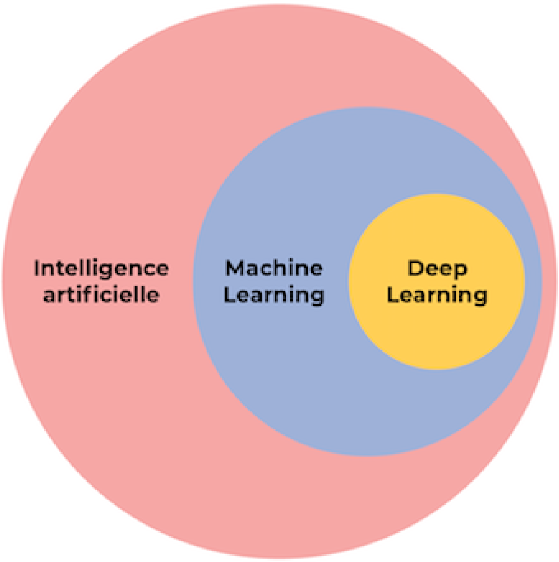


Figure I.8 Relationship between artificial intelligence, machine learning, and deep learning [45]

6. Domains of AI:

The field of AI contains several specialized subdomains, each focusing on distinct aspects of human intelligence simulation. Principal subdomains include:

6.1. Robotics:

This branch of AI focuses on developing autonomous systems capable of intelligently interacting with their environment. It integrates:

- Sensor perception (processing environmental inputs).
- Path planning (navigation decision-making).
- Motor control (Physics movement execution).
- Real-time decision-making (dynamic response to changes).

Robotics combines hardware and AI algorithms to create adaptive, goal-driven machines [4][8].

Figure I.9 AI in robotics [46]

6.2. Expert system:

Expert systems (ES) are a branch of applied artificial intelligence (AI), and were developed by the AI community. The basic idea behind ES is simply that expertise, which is the vast body of task-specific knowledge, is transferred from a human to a computer. This knowledge is then stored in the computer and user, but they don't include many possible situations [8].

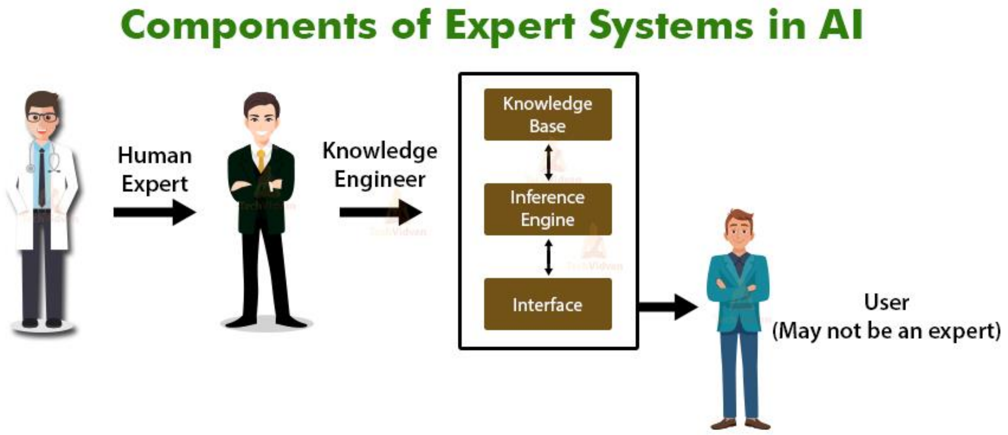


Figure I.10 How expert systems function [47]

6.3. Natural Language Processing (NLP):

Natural Language Processing (NLP) is a part of artificial intelligence that helps computers understand and work with human language. It combines understanding language and using computers to do tasks such as organizing text, translating languages, and making chatbots [2][6]. NLP integrates:

- Computational Linguistics (rule-based modeling of human language) [6].
- Statistical models [6].
- Machine Learning [6].
- Deep Learning [6].

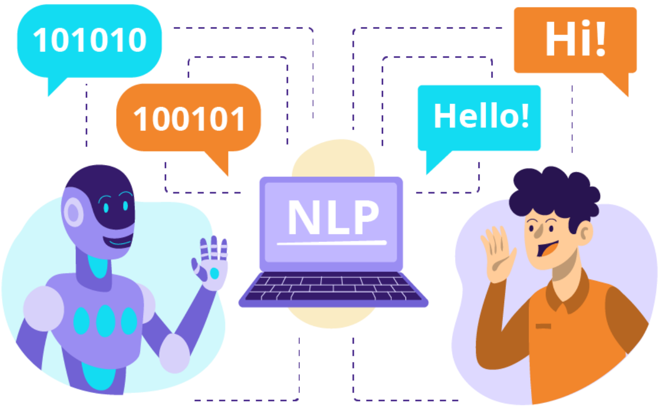


Figure I.11 Natural language processing in AI [48]

6.3.1. Large Language Models (LLMs):

Large Language Models (LLMs) are advanced AI systems that are trained on big collections of text to understand and create language that sounds like what humans speak. Using transformer models, they perform better than expected in tasks like finishing text, understanding information, and responding to questions. Today’s LLMs, like Llama 3, use tools and self-learning methods to prepare effectively. These models can work with long documents (like PDFs) and have larger setting limits (for example, 128,000 words in Llama 3). Main uses include summarizing documents, chatbots that can have conversations, and a technology called retrieval-augmented generation (RAG). Unlike older versions, Llama 3 has improved support for multiple languages (including Arabic and French) and is more efficient, making it suitable for organizations with fewer resources. In any case, there are still challenges in reducing bias, managing mental processes, and handling complicated reports quickly [2].

6.3.1.1. Large Language Model Families:

Huge Bunches of Dialect Models Cutting edge dialect models can be assembled into three primary sorts based on how they work:

1. Autoregressive models: (like Llama 3 and GPT-3.5) make content one word at a time, which is awesome for chatbots and inventive composing [2][18].
2. Bidirectional models (like BERT and RoBERTa) see at the encompassing setting from both bearings, which makes them great at assignments like categorizing data and distinguishing names [2].
3. Multimodal PTMs (like GPT-4V and Flamingo) can work with both text and visual inputs, making them to better understand documents that include textual and graphical elements [19].

Autoregressive models are prevalent in conversational AI since they sound smooth and characteristic. On the other hand, bidirectional models are way better for understanding the meaning of content. Multimodal LLMs are being utilized increasingly to work with PDFs that have both text and pictures. The choice of a demonstrate depends on what you would like it to do, how much computer control you have, got and what dialects it ought to bolster [2][18][19].

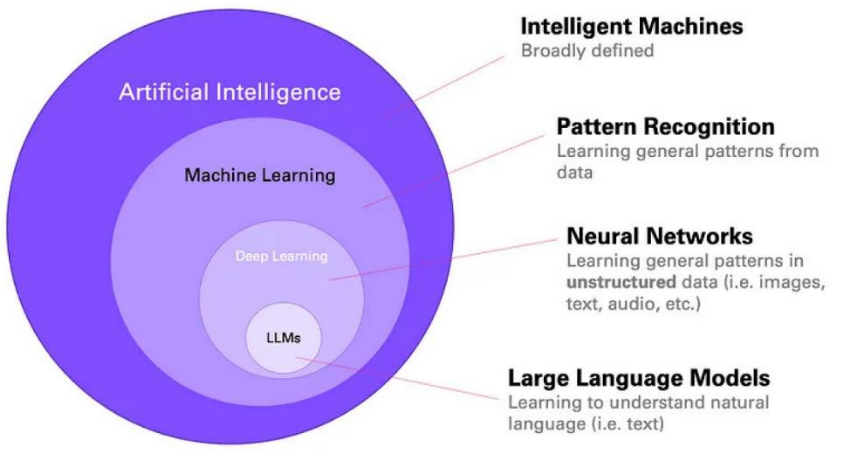

Artificial Intelligence
Machine Learning
Deep Learning
LLMs
Intelligent Machines
Broadly defined
Pattern Recognition
Learning general patterns from data
Neural Networks
Learning general patterns in **unstructured** data (i.e. images, text, audio, etc.)
Large Language Models
Learning to understand natural language (i.e. text)

Figure I.12 Relationship between artificial intelligence, machine learning, deep learning, and large language models [49]

6.3.1.2. LLAMA 3:

Llama 3, created by Meta AI, is a big upgrade in open-weight language models and performs better than the earlier versions. It can handle long documents, like PDFs, because it can use up to 128,000 tokens, and its special way of focusing allows it to work faster. Llama 3 has learned from 15 trillion words in different languages. It works great in Arabic and French, so it's good to use in Algeria. The model comes in different sizes, from 8 billion to 70 billion parts. This means it can perform well with different amounts of resources. Special versions like Llama 3-Instruct are really good at talking and can help make chatbots that communicate well. Its open-weight design makes it cheaper than private options, but it can still be hard to use well on devices with limited resources [2][12].

6.3.1.3. Possible scenarios for using LLMs:

Large Language Models (LLMs) like Llama 3 have improved many fields because they are really good with language. In document processing, they help to quickly find important information, summarize details, and answer questions from complex PDFs and reports. This directly helps with looking at legal agreements and doing school research. Large Language Models (LLMs) help make smart chatbots that can talk in a way that fits the situation. They can make customer service easier and serve as personal helpers for learning. In healthcare, they assist by reviewing medical documents and patient records. In software development, they help create programs and solve issues. LLMs are good at handling various languages. They help break down language barriers by offering great translation and localization services, which is really helpful in areas with many languages, like Algeria. Their link to retrieval-augmented generation (RAG) systems helps them be more accurate because they can combine what they create with information from outside sources. There are still issues with how quickly computers can run, so we need to make them better for everyday use. We need to consider being fair and taking care of the content we share. Future improvements might focus on making these tools fit better in different areas and helping them work in real time so they are more helpful in various industries [2][8].

7. Conclusion:

This chapter looked at the basic ideas of Artificial Intelligence, how it has developed over time, and how it is changing the way we use technology, especially in processing text and handling documents. We looked at important parts of AI, such as Machine Learning, Deep Learning, and Natural Language Processing (NLP). We also studied advanced systems like Large Language Models (LLMs) and how they can work with unstructured data. The discussion talked about how AI is used in different areas, like robots and systems that provide expert advice. It also mentioned problems such as how accurate AI is and ethical issues. As AI gets better, models like Llama 3 show how machines can change the way they understand and use human language. These changes prepare us for more discussions in the next chapters, where we will explore how to use and solve problems with AI for processing textual data such as PDF files.

Chapter II: ChatBot

Chapter II: ChatBot

1. Introduction:

In recent years, chatbots have become a useful and adaptable technology that has changed the way people use digital systems. These intelligent chatbots can hold conversations like humans. They enable users to request information, complete tasks, or get help in everyday language. Chatbots are now more advanced thanks to the rapid development of artificial intelligence and natural language processing. You can have complex conversations instead of just giving pre-written answers [2].

Chatbots have a long history, starting with simple programs like ELIZA. They have changed over time, with important steps including ALICE and IBM Watson, leading up to today's AI helpers like Siri and ChatGPT. Each stage made it better at understanding language, personalizing for users, and working in different areas. Today, chatbots are used in customer service, learning tools, and smart helpers, showing how important this technology is in our online lives [2][14].

Modern chatbots use techniques to understand language better. These include figuring out feelings in messages, breaking down sentences into individual words, fixing text to make it clearer, and identifying the roles of words in sentences. This helps them understand what users want and give correct answers. There are different types of systems, including rule-based systems, retrieval-based models, and generative models like Sequence to Sequence. These new features allow chatbots to put together summaries, get important details, and answer questions using documents shared by users, like PDF files [2][14].

This chapter looks at the basics and parts of chatbot technology, especially how it works with language processing and handling documents. It also gives a clear overview of different types of chatbots, how they are built, the tools and systems used to create them, and the problems that can come up. A look at current studies and developments shows that chatbots are being used more and more in different areas. This highlights the importance and uniqueness of our own project in this growing field.

2. Chatbot:

2.1. Definition:

A chatbot, also known as a conversational agent, is a computer program that is made to have conversations like a human, using text or speech. In today's online world, this technology is becoming more common, especially in customer service for businesses and online shops using instant messaging [2][15].

These smart systems are built to help users with different tasks quickly and effectively, such as finding information, managing reservations, processing orders, and answering general questions. Chatbot technology can be used in many ways and keeps getting better as artificial intelligence (AI) improves [2].

Chatbots can be powered by predefined scripts or utilize more advanced technologies such as natural language processing (NLP) and machine learning to understand and respond to user inputs more effectively. They are commonly used in customer service, virtual assistants, processing textual data, and various application scenarios to enhance user experience and streamline operations. The program's functional mechanisms involve the ability to process incoming messages such as emails, SMS, and chat messages and generate appropriate responses within a matter of seconds. This rapid response capability is crucial for maintaining efficient communication and providing timely assistance. Additionally, the program can utilize various algorithms and natural language processing techniques to understand the context of the messages and tailor its replies accordingly. This functionality enhances user experience and ensures that inquiries are addressed promptly and accurately [2][15]. You can use:

- Ready-made answer templates or [2]
- Get help for more advanced methods to create natural language with better tools [2].

Chatbots have improved a lot over the years. They have changed from rule-based systems that follow simple rules to advanced AI models, which shows how much our way of interacting with computers has improved [14].

2.2. Chatbots History:

Talking tools like bots and chatbots have changed the way people interact with computers by making communication easier and more natural. The history of these systems shows great improvements in technology.

The field started in 1966 when Joseph Weizenbaum created ELIZA at MIT. It was the first program that could imitate human conversation by matching patterns and using pre-written replies. This groundbreaking work set up the main structure for future systems [14].

In 1972, Kenneth Colby made a program called PARRY at Stanford University that was an important step forward because it imitated paranoid schizophrenia. This system was better at managing conversations than ELIZA [14].

In 1992, Dr. Sbaitso did this next. Sbaitso was a computer program for MS-DOS that acted like a psychologist by talking to users with a computer-generated voice [20].

Richard Wallace's ALICE (1995) was an important development. It used pattern-matching techniques and was built using AIML (Artificial Intelligence Markup Language) to simulate conversation. This system won the Loebner Prize three times for its improvements in chatting with people [14].

In 2001, SmarterChild showed that smart chatbots could be useful and profitable for instant messaging apps [13]. offering:

- Finding information right away [13].
- Functions of a dictionary and encyclopedia [13].
- Simple entertainment options [13].

IBM's Watson (2006) was a big step forward in smart computing [13]. It could:

- Understanding natural language [13].
- Rapid retrieval of relevant information [13].

Its win on Jeopardy in 2011 showed amazing skills in understanding messy information [13].

The current time started with Apple's Siri in 2010 [20], which combined:

- Natural language interaction through text and voice [20].
- Understanding human language [20].
- Integration with apps and messaging for responsive digital assistance [20].

Later events included:

1. Google Now (2012): Understanding context through:
   - Time and location data [20].
   - Helpful predictions [20].
2. Amazon Alexa (2014):
   - A way to use your voice to interact with devices [20].
   - Skills-based architecture refers to a way of organizing things based on the specific skills or abilities needed for different tasks [20].
   - Connecting smart devices in your home [20].
3. Microsoft Cortana (2014): Focus on assistant utilizing:
   - Connecting different platforms or systems so they can work as one [20].
   - Helpful ideas [20].

In 2016, Facebook launched its Messenger Platform, which allowed anyone to create chatbots. Within a few months, there were 11,000 chatbots available [13][21]. This environment allowed for:

- Automated customer service means getting help from machines or computer programs instead of talking to a person [21].
- Online shopping orders [21].
- Sending tailored content to individuals.

The Transformer Revolution (2017-2019) refers to a big change in technology, focusing on a new model called Transformer. This model improved how machines understand and process language. The launch of Transformer models changed the way we think about conversational AI [2][22], allowing for:

- Attention methods to help understand text based on context [2].
- Training language models in a way that can grow easily [2].
- Understanding context from both directions (like BERT) [2][22].

These improvements helped create chatbots that understand documents better by enhancing how they analyze both organized and unorganized text [2].

Retrieval-Augmented Generation (2020-2021) is a method that helps computers find and use information better when creating text [2][23]. The creation of RAG frameworks solved important problems in pure generative models by:

- Using neural search together with generative AI [2].
- Allowing accurate answers from collections of documents (like PDF files) [2].
- Providing answers that are backed by evidence for school or legal purposes [2].

Long-Context and Multimodal Systems (2022-2023):

This refers to systems that can understand and use a lot of information over a long time and can work with different types of data, like text, images, or sound. Improvements in increasing context window size (like having 128,000 tokens in GPT-4 Turbo) made it possible to [22][24]:

- Complete handling of long documents (like research papers and contracts).
- Comparing and combining information from different documents.
- Combining different types of information (like text and tables in PDFs).

Focusing on specific skills and making things more accessible to everyone [22].

(2023-Present) Recent improvements have been centered on:

- Improving processes for specific areas (like understanding legal or medical documents) [22].
- Open-weight models (like Llama 3) allow for cheaper use [22].
- Full process tools (like LangChain) make development easier [25].

2.3. Types of chatbots:

- **Rule-Based Chatbot:** The chatbot replies to customer questions using set rules. Making a rule-based chatbot is simple and inexpensive [14].
- **Retrieval-Based Models:** Retrieval-based chatbots use graphs to find and share information. The chatbot is designed to give the best answer it can. The answers come from the information that is already available. Retrieval-based chatbots use methods like matching keywords and machine learning. Using deep learning to find the best answer. These chatbots only give set answers and do not create new replies [14].
- **Generative models:** Are types of neural networks that create new responses instead of using pre-made answers like retrieval-based systems do. They create new answers from the beginning. Answers come only from machine learning and the data it learned from. Sequence to sequence models. Here’s a simpler version. Chatbots that use generative methods can create new conversations by learning from a lot of previous chat examples [14].

2.4. NLP in Chatbots:

2.4.1. Translation:

The first use of natural language processing was in machine translation, which is used to translate one human language to another. It helps users to understand foreign documents and translate a text from one language to another, for example, Google Translate [2].

2.4.2. Sentiment analysis:

Sentiment Analysis is a way to use natural language processing and statistics to figure out how individuals feel or what they think about something. It used to automatically understand what people think, feel, and want from what they write. This strategy is used in various situations. It is used on the internet to determine if someone feels positive, negative, or neutral based on the input text. It is often used to assist businesses in understanding how customers feel about their brand and products by looking at feedback and to learn what clients need [2].

2.4.3. Question answering:

A question answering (QA) system is made to provide answers to questions in any human language using the information found on the documents that have been uploaded. It knows what you mean and only provides the information you ask for, and can even provide answers at the level of user understanding, unlike search engines [2].

2.4.4. Tokenization:

Tokenization is a basic method used in natural language processing. It splits long pieces of text into smaller parts called chunks or tokens after taking out punctuation or special characters. The main goal of tokenization is to help understand and make sense of the text. Tokenization can be used to break up text into individual words or sentences. In simple terms, big pieces of text can be broken down into sentences, and sentences can be broken down into words [2].

- **Sentence Tokenization:** Breaking text into individual sentences. Tokenization of sentences is the way of breaking text into clear and complete sentences. This seems like a really easy idea. In languages like English and French, we can divide sentences whenever we see a punctuation mark. But because we use periods for abbreviations (for example, Dr. Benali), this issue is not simple, even in English. When working with regular text, lists of abbreviations with periods can help us avoid misplacing where sentences end. Often, we use libraries to help us with this, so don't worry about the details for now [2].
- **Word Tokenization:** Breaking text into individual words. Like the process mentioned earlier, where we pick out each sentence from a text, word tokenization is when we break a sentence into individual words or parts. You can change the word tokenization results into a Data Frame to make it easier to understand text in NLP applications [2].

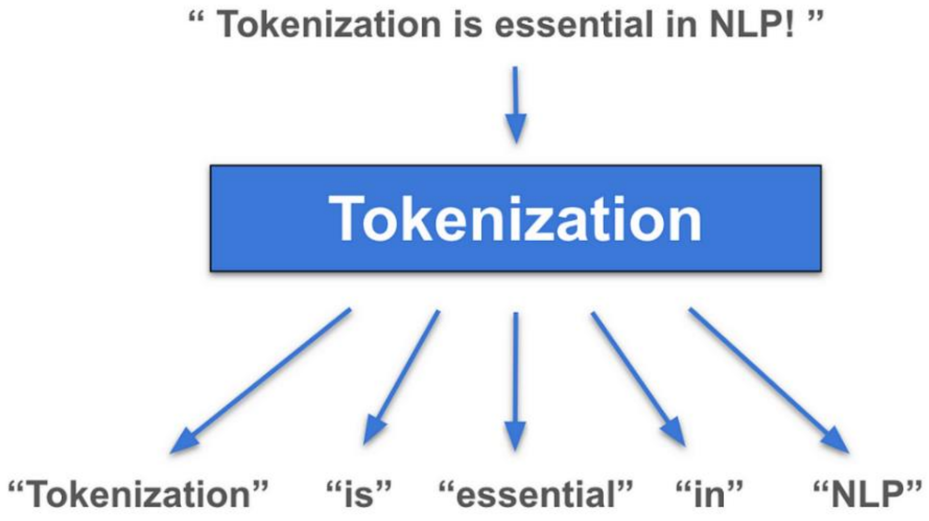


Figure II.1 Example of tokenization [50]

2.4.5. Normalization:

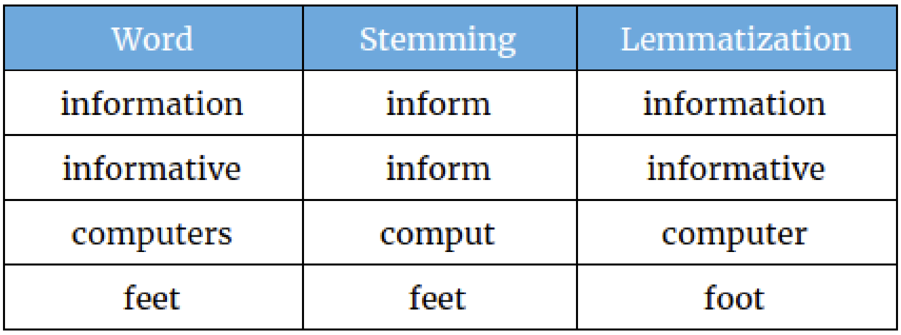
Normalization is an important process in natural language processing (NLP). It's a way to make text better and more organized by getting rid of different forms of words. In other words, it changes a token into its basic form. This makes it easier for the computer to handle less information, which helps it work better. Two main methods for normalization are lemmatization and stemming [2].

- **Lemmatization:**

Lemmatization is a key method in natural language processing (NLP) that helps prepare text for analysis. It has a goal. to put different versions of a word together by changing them to their simplest form. This root the word "lemma" is the basic form of a word that you would find in a dictionary. for example, the word "loves" converts to "love". The only difference between Lemmatization and Stemming is that Lemmatization does it in a proper way, it can use a dictionary [2].

- **Stemming:**

Stemming is a basic method for making text simpler in natural language processing (NLP). It's the process of removing parts of a word (like prefixes and suffixes) to get to the base form of the word, without considering its meaning, for example, the word “loved” stemmed to “love” [2].



Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

Figure II.2 Stemming and Lemmatization: reducing words to base forms for NLP [51]

2.4.6. Part Of Speech tagging (POS):

Part of speech (POS) tagging is the way of marking words to show what role they play in a sentence, like whether they are a noun, verb, or adjective. This is done based on what the words mean and how they are used in the sentence. POS tagging is an important task in natural language processing that helps analyze sentences and understand texts better by showing how words are used in relation to each other. POS tagging labels each word in a text with a category like NOUN, VERB, or ADJ. This changes plain text into a form that can be better understood and analyzed by computers. For example, classifies "How" as a question word that shows we are asking something [2].

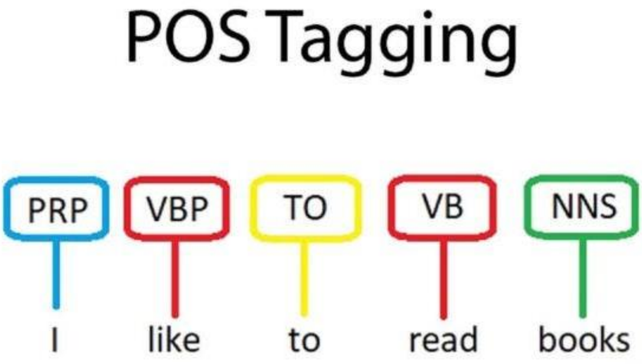


Figure II.3 Example of POS tagging [52]

2.4.7. Word Embeddings:

Word embeddings are a foundational component in modern Natural Language Processing, providing dense vector representations of words that capture their semantic relationships. Embeddings map words into continuous vector spaces where similar meanings are positioned closer together. This representation allows chatbots to better understand the context and intent behind user input. Models such as Word2Vec, and more recently, contextual embeddings like BERT, have significantly enhanced the performance of chatbot systems. To efficiently retrieve semantically similar responses or documents, libraries such as FAISS can be used alongside embedding models. Embeddings enable the system to detect nuances, synonyms, and relationships between words, improving both intent classification and entity recognition. In chatbot pipelines, these vectors are typically fed into deeper architectures such as Transformers.

This allows for more accurate response generation and dialogue management. The use of pre-trained embeddings also reduces the need for large datasets, enabling efficient development. Overall, embeddings are critical in bridging the gap between human language and machine understanding in chatbot applications [2][39].

2.4.8. Natural Language Understanding (NLU):

The first important part of Natural Language Processing (NLP) is Natural Language Understanding (NLU). This helps computers understand and make sense of what people say or write. This technology helps a chatbot understand what users are asking and what they really mean. For example, let's say a user talks to a chatbot by asking: "How can I get a book to read." The NLU system handles this question by:

Intent Recognition: Figuring out what the user wants to do (like asking about borrowing a book).

Entity Extraction: Finding important things (like "book" as what we're looking at).

Contextual Analysis: Clearing up confusion (for example, telling the difference between borrowing something physical and borrowing something digital).

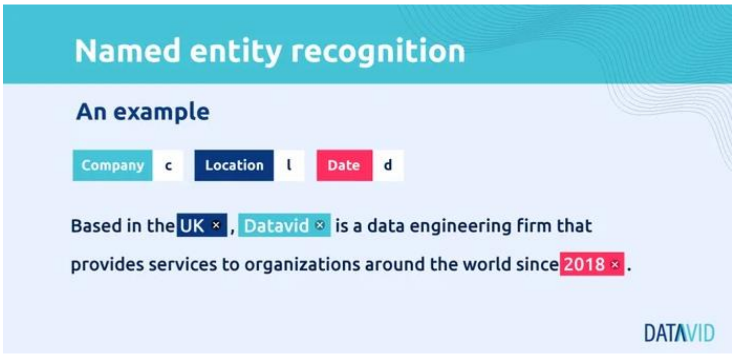
Without strong natural language understanding (NLU), chatbots can only match keywords, which makes them less useful for complicated tasks like processing documents or communicating in different languages [2][14].

2.4.9. Natural Language Generation (NLG):

Natural language generation is a part of understanding and using human language with computers. natural language understanding helps computers understand what they read, and natural language generation helps them write. NLG is the way computers create human-like language using information from documents or questions from users. natural language generation can make short summaries of documents while making sure the main information is correct [2][14].

2.4.10. Named Entity Recognition (NER):

NER is a process that finds and sorts names of people, places, or things from written text that doesn't have a specific structure. Named entities are real things that we usually name with specific words, like names of people, places, or organizations. NER is used to take out names and important words from the text before checking how people feel about it, and to give fixed scores to sentences. Removing specific names or proper nouns helps make extractive text summarization better [2].



Named entity recognition

An example

Based in the UK ×, Datavid × is a data engineering firm that provides services to organizations around the world since 2018 ×.

Figure II.4 NER example [53]

2.4.11. Stop-word removal:

In natural language processing (NLP), stop words that don’t add much meaning are usually taken out when preparing the text. These include words like articles (the, a, an), prepositions (to, in, on), conjunctions (and, but, or), and helping verbs (be, have), among others. Even though they appear a lot in texts, their commonness means they aren't very helpful for things like finding information or analyzing feelings [2].

Example in French language: le, la, les, ainsi, toutefois…

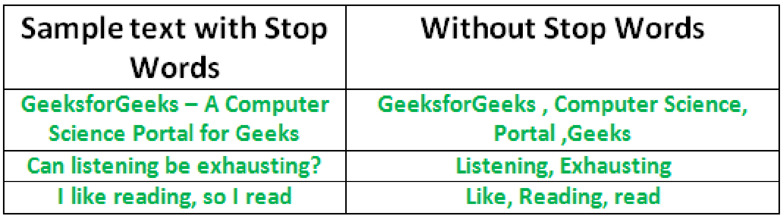
Example in English language: the, and, about, have, can…

These words should be taken out of written texts for two main reasons:

Language Viewpoint: Stop words carry very little useful information. Their presence or absence doesn't really help us understand the meaning of a text [2].

Looking at it from a numbers point of view: Stop words are found in all texts equally and without bias. Because they don't have enough unique qualities, they don't give helpful information for tasks like sorting texts or grouping them [2].

By removing common words that don't add much meaning, we make the data cleaner, help our computers work faster, and ensure that the important words left are more useful for later tasks in natural language processing [2].



Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

Figure II.5 Example of stop words removal [54]

2.4.12. Sequence to sequence models:

These models are designed to handle tasks where the input and the output have different lengths like machine translation and summarization. Sequence to sequence consists of an encoder, which processes the input text, and a decoder, which generates the output.

Sequence to sequence models can be seen as a system where one decoder part makes one output that matches one encoder part. This model is easy to use for translating languages because the decoder can create a word in the output language by looking at one word from the input language at a time [2].

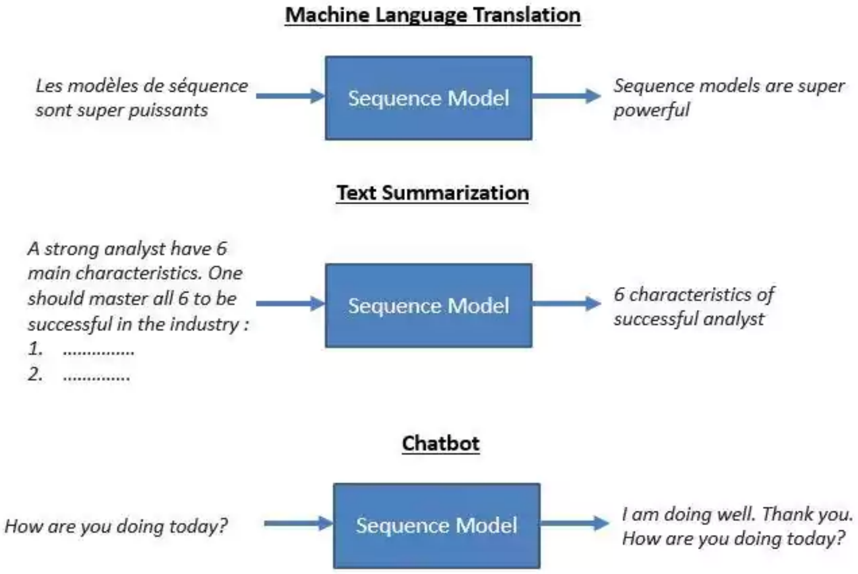

Machine Language Translation
Les modèles de séquence sont super puissants
Sequence Model
Sequence models are super powerful
Text Summarization
A strong analyst have 6 main characteristics. One should master all 6 to be successful in the industry :
1. ....
2. ....
Sequence Model
6 characteristics of successful analyst
Chatbot
How are you doing today?
Sequence Model
I am doing well. Thank you. How are you doing today?

Figure II.6 Sequence to sequence models in chatbots: powering text summarization, translation, and managing dialogue [55]

2.5. Open domains and closed domains Chatbots:

2.5.1. Open domain:

Open domain chatbots are chat systems that can talk about many different subjects without being limited to any one topic or purpose. Unlike bots that have specific jobs like helping customers or making bookings, open domain bots are designed to have natural conversations, just like chatting with a friend. They should be able to understand different types of questions and give suitable answers on many topics, ranging from fun things like movies and hobbies to technical subjects. Open domain bots can be easier to start because there are lots of public data available, like social media chats and movie scripts. However, they also come with big difficulties. This includes needing a lot of general knowledge, keeping talks clear and connected over long chats, and dealing with surprise responses from users. One well-known example is ChatGPT, which is made to talk about almost any topic in a very relevant and smooth way [14][15].

2.5.2. Closed domain:

Closed domain chatbots are computer programs that can talk about a specific subject or perform a particular task. They fix specific problems and help users reach their goals more easily, like shopping assistants, booking a ticket, or getting help with tech issues. These systems have a fixed way of communicating and usually have a few specific questions and answers for the business or service they support. closed domain bots need a person to make the final decisions because their conversation paths are planned out in advance. Even if users try to talk about something else, the system can't understand or provide useful answers to those queries. Their main job is to complete certain tasks as efficiently as they can [14][15].

2.6. Architecture components:

The chatbot's design shows how its main parts work together. These parts help the system understand what users say, handle PDF files, and create fitting replies. Each part is made to do a specific job in the conversation process, helping the chatbot have natural and smart talks using information from documents.

2.6.1. User interface (UI):

We built a web-based chatbot, focusing on designing a friendly user interface, that enables users to use our chatbot smoothly and easily, where the user can upload one PDF or multiple PDFs in particularly any field, and have a conversation with the PDF uploaded in natural language. We focused on the UI to ensure a smooth interaction with the backend and to provide a friendly user interface.

2.6.2. Document processing module:

This component is responsible for parsing and preprocessing the content of uploaded PDFs. Using tools like PyMuPDF, the system extracts text while handling different layouts and structures. Preprocessing steps include cleaning the text, segmenting paragraphs, and preparing the content for analysis [2].

2.6.3. Preprocessing:

Preprocessing a text means taking several steps to organize a document so that it is easy to understand and analyze. In any text, some words and symbols don’t have much meaning connected to the topic. They are usually used to join other words together. If these words keep showing up, they can affect the score of the important words. To fix this problem, people have used different methods such as breaking text into smaller parts (Tokenization), making words standard (Normalization), removing common words (stop word removal), and so on [2][6].

2.6.4. Understanding Intent and Directing Tasks:

When a user asks something, the system figures out what the user wants, like if they want a summary, to get certain information, or to ask a question. According to this plan, the request is sent to the right part (like a summarizing tool, information gathering tool, or question and answer section). This process depends on the Natural Language Understanding (NLU) parts we talked about earlier.

2.6.5. Specific task module:

Depending on the user query, the system will choose one of these:

- **Summarization:** Not all information in a text document is helpful, so unnecessary parts can be removed, and just the important stuff should be read. Summarization means making a document shorter by including only the important information and leaving out the extra details. For example, rather than reading a long news story, you can use summarization to get just the important facts about the news event. You could make a short summary of the article to help you decide if you want to read the whole thing. We were able to make our chatbot generate a concise summary of the uploaded document using pre-trained models.
- **Information extraction:** Another key task in natural language processing (NLP) is to pull out helpful information from documents for various uses. For example, companies that are publicly traded must share their financial information with their shareholders. NLP can help pull out financial details from these documents to automatically collect information about how a company or industry is performing. If gathered carefully, the information from these sources can help companies make good choices. For example, if a company didn't do well last quarter, you could use NLP to find this out and then automatically sell your shares of their stock [2].
- **PDF Q&A:** The Question Answering (Q&A) functionality is a central feature of the chatbot, allowing users to ask questions in natural language and receive instant, accurate answers based on the content of an uploaded PDF document. When a user submits a query, the system analyzes the question, retrieves the most relevant parts of the document, and generates a human-like response. This creates a conversational experience that feels more dynamic and personalized compared to a traditional static FAQ page. The main advantage of this Q&A system is its flexibility and availability. The chatbot can handle a wide variety of questions, guide users to specific content within the PDF, and provide consistent support at any time without the limitations of predefined question sets or human availability. This makes it an efficient solution for assisting users in understanding complex documents, learning about products or services, or extracting key information quickly [2].

2.6.6. Response delivery:

The Response Generator is an important part of the chatbot system. It creates the final message that the user sees. After the system finds or creates the needed information, like a summary, specific data, or an answer, this part changes the raw result into a clear and correct sentence or paragraph that makes sense. This part usually uses Natural Language Generation (NLG) methods, often supported by advanced language models like GPT-3.5 or Llama 3. These models can create answers that sound like a real person and fit well with what the user asked, making it feel like a regular conversation. The Response Generator looks at the message content, what the user wants, past conversations, and how the language is used to make sure the reply is clear and appropriate [2]. A powerful response generator makes things better:

- Ensuring the answer is simple and clear.
- Keeping users interested by having natural conversations.
- Giving answers that seem trustworthy and like a real person.

2.7. Frameworks and platforms for building chatbots:

Actually, building chatbots has become much easier because we have in hand many frameworks and platforms that offer tools, libraries, and services. These helps make it simpler to design, train, and launch smart chatbots. These tools help understand human language, manage conversations, and connect with user interfaces. This lets developers create chatbots that focus on what users need and the tasks they want to complete, without having to start from zero.

2.7.1. Overview of common chatbot frameworks and tools:

To implement a web-based chatbot, there are many tools and frameworks that enable developer to realize it, list of the most used ones:

Botpress:

- Modular, open source chatbot framework.
- Built with Node.js, has visual tools for flow building.
- Suited for developers who want more control without starting from zero.

Rasa:

- Open-source framework based on Python [27].
- Offers training pipelines, natural language understanding, dialogue management, and Rasa x for testing [28].
- Great for flexibility and data privacy [27].

IBM Watson assistant:

- Cloud-based service to create chatbots with NLU and machine learning [6].
- Works smoothly with business systems [6].
- offers a visual dialog editor [6].

LangChain:

- Open-source framework [25].
- Enable chaining of LLM operations and manage memory [25].
- Allows programmers to manage interactions between the PDF content, the user’s questions, and the LLM [25].

Streamlit:

- Open-source Python web-based framework [29].
- Enable fast development of a responsive and interactive front end for users [29].
- Easy to use [29].

Python:

- High-level programming language [30].
- Strong ecosystem [30].
- Enable building web applications and mobile applications [30].
- Contains many libraries that help developers build their applications [30].

ChatGPT API (OpenAI):

- offer access to powerful LLMs such as GPT-3.5 and GPT-4 [22].
- Allows developers to use GPT (Generative Pre-Trained) models for generative chatbots.
- Simple to integrate [22].
- Very powerful for open-domain Question answering and summarization tasks [22].

2.7.2. Open-source and Cloud-based:

There are two main types of chatbot frameworks:

- **Open-source frameworks:**
  These are frameworks for developers that allow them to completely manage how the chatbot works, how it learns, and where it runs. Some examples are Rasa, Botpress, and LangChain. These tools are great for projects that need special rules, privacy, or the ability to work offline [22].
- **Cloud-based platforms:**
  These are online setups that offer pre built APIs, services, and tools for making chatbots. Some examples are IBM Watson Assistant and OpenAI's ChatGPT API. They are easy to use, can grow as needed, and need less need for maintenance, which makes them good for quick development and launch [26].

The choice between these options depends on the specific project requirements, budget, customization needs, and hosting preferences.

2.8. Common limitations and obstacles:

2.8.1. Security:

There are risks when using chatbots, such as people denying their actions, pretending to be someone else, changing information, revealing confidential data, overwhelming the system, and gaining unauthorized access to higher permissions. A chatbot can be at risk of attacks if it is not kept up to date, not programmed well, or doesn't have enough security. It's important to be careful about security with chatbots because they can pose new risks. Companies need to improve their security to protect users information [2][14].

2.8.2. Context Integration:

It’s really important for chatbots to understand what people mean in their replies. The first challenge to overcome is adding context to the chatbot. To create sensible answers, we need to consider both the physical environment and the language used. To include language context, conversations are put into a vector, which makes things difficult. The chatbot needs to include information like where users are, when they are chatting, their relationships, and details about them [14].

2.8.3. Harmful content:

A big problem for chatbot makers and users is harmful content. For example, harmful content can come from unreliable services that collect personal information, or from comments meant to misuse chatbots, violate privacy, or steal copyright. Chatbots need safety measures to prevent them from being used wrongly [2].

2.8.4. Exit Plan:

Chatbots aren't very good at having a conversation. They have a fixed number of responses and reactions. When a chatbot can't help a user with their problem, it should have a backup option [14].

2.8.5. Goals and Variety:

A usual issue with generative systems is that they often give simple responses like "That's interesting" or "I don't know" which can fit many situations. That's partly because of how these systems are taught, including the information they use and the goals they are given during training. Some researchers have tried to increase diversity in a planned way using different methods. However, people usually give answers that are directly related to what they received and have a purpose. Generative systems, especially those that can talk about any topic (open domains), don't have specific goals in their training. Because of this, they don't show a lot of variety [15].

2.8.6. Checking Models:

The best way to see if a chat program is good is to watch how well it solves a customer support problem during a chat. But getting these labels costs a lot because it needs people to make decisions and evaluations. Sometimes there isn't a specific goal, especially with models that can handle many different topics. Common metrics like BLEU, which are used to measure machine translation, work by comparing text. However, they don't work well because good responses can use different words or phrases [15].

2.8.7. Easy to use:

A good chatbot should be simple, clear, and friendly for users. People should be able to use it without needing any training. However, users might get annoyed if the chatbot doesn't understand what they mean, gives confusing answers, or asks for very specific types of information. Creating user-friendly interfaces that help users easily understand how to use them, manage mistakes smoothly, and keep conversations flowing is very important, but it can be hard to do. Also, making sure to meet different user preferences, languages, and accessibility needs makes things more complicated. A bad user experience can make people less interested, unhappy, and might cause them to stop using the system.

3. Overview of existing research and developments:

- A chatbot for customer service for the Algerian railway company (SNTF) using Google Dialogflow. The chatbot helps users book tickets, check train times, and answer general questions through a chat. Dialogflow has tools that help it understand language and work in the cloud, making it easy and quick to build and connect new features. The bot uses goals, keywords, and steps to correctly respond to user requests. Firebase and Firestore were used to manage the database in real-time and provide support for the backend. The system is available all the time, making it easier and quicker for customers to get help. You can reach the chatbot through different places, like websites and messaging apps. Dialogflow lets you have flexible and more natural conversations. The chatbot was tested and confirmed to work well in real situations [15].
- A smart chatbot to help with customer service at Algeria Telecom. The system uses AIML (a language for artificial intelligence) to handle answers and update information. It uses facial recognition and voice-to-text technology to make customer interactions more personal and communication easier. The chatbot can tell if a customer has been

here before, adjust its answers based on how the customer feels and their facial expressions, and reply using both voice and text. Information is regularly updated using APIs and by gathering data from Wikipedia and the Algeria Telecom website. The system has parts that can find similarities using cosine similarity, connect with devices like a camera, microphone, speaker, and screen, and analyze emotions in real-time. The chatbot was made using C#, SQL Server, and EmguCV. It wants to make things easier for reception staff and help them connect better with customers [16].

- The study presents a conversational PDF chatbot that combines Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs) to enable natural language interactions with uploaded documents. The system processes PDFs using PyPDF2. The chatbot, deployed via a Streamlit interface, highlights source passages in the PDF and supports multi-turn dialogues. Outperforming baselines in extracting coherent answers. Unlike generic chatbots, it focuses exclusively on user-uploaded documents, ensuring privacy and relevance [17].

4. Conclusion:

In conclusion, this chapter has examined the development of chatbot from a simple rule-based system to a more intelligent and AI-powered conversational agent. We looked at necessary NLP techniques that allow chatbots to be able to analyse and understand human languages. These are sentiment analysis, question answering, normalization and named entity recognition. We also showed the difference between open and closed domain bots, along with their use cases. We showed the architecture and components of a document aware bot that needs modules for document processing, intent recognition, and response generating. A review of key frameworks like LangChain, Streamlit, and GPT-based models shows what modern frameworks can do. In the end we communicated limitations of the system and discussed several other research projects. Thus, demonstrating how important chatbots are growing in terms of document and customer service. This groundwork prepares the reader for the in-depth design and construction described in the following chapter.

Chapter III: Deployment and Results

Chapter III: Deployment and Results

1. Introduction:

Chapter 3 presents the modeling and implementation aspects of our proposed chatbot system. We initially regard the modeling activities such as the analysis and design of the system using UML, which is the foremost means of visualizing both the static and dynamic aspects of a system. This modeling specifies all structural and behavioral aspects of the chatbot and directs the implementation of the subsequent phase.

Subsequent to the modeling step, we engage in the technological implementation, wherein we showcase the chatbot's efficacy in user interaction and delivering pertinent replies. We provide comprehensive insights into the chatbot's operational functionality by developing a prototype that validates the dependability and performance of our suggested conversational solution.

2. Architecture of the proposed approach:

2.1. Modeling Instruments and Languages:

Modeling tools and languages denote software programs and formal languages intended to aid in the construction, visualization, and analysis of abstract representations of systems, processes, or data structures in computer science. These tools and languages empower computer scientists, engineers, and researchers to envision, build, and simulate intricate systems or algorithms, facilitating the creation, comprehension, and optimization of software and hardware solutions. They often provide graphical interfaces, specific notations, and simulation functionalities to depict and examine diverse elements of computer systems, including software architectures, algorithms, data structures, and communication protocols [31].

In the realm of engineering and research, several modeling tools and languages are available, each designed for distinct requirements and domains. Some of the most used tools and languages are presented as follows [31]:

The Unified Modeling Language (UML) is extensively used in software engineering, offering a standardized method for visualizing system architecture. It provides several diagram kinds, including class diagrams, sequence diagrams, and activity diagrams [31].

2.2. Unified Modeling Language Overview:

Unified Modeling Language (UML) is a standardized visual modeling language used in software engineering for the specification, visualization, construction, and documentation of software-intensive systems. UML provides a set of graphical notation techniques to represent system structures, behaviors, and interactions in a standardized manner, facilitating communication and understanding among stakeholders involved in software development projects [31].

The UML diagrams are diverse and capture both the static and dynamic aspects of a system. Below is a list of UML's most used list of diagrams: [31]

- **Class Diagram:** Represents the static structure of a system by showing classes, their attributes, methods, and relationships between them.
- **Use Case Diagram:** Describes the interactions between a system and its users or external systems, illustrating the functional requirements of the system.
- **Sequence Diagram:** Shows how objects interact in a particular scenario over time, depicting the sequence of messages exchanged between objects.
- **Activity Diagram:** Illustrates the flow of control within a system, depicting the sequence of activities or actions performed by system components.
- **Component Diagram:** Depicts the physical components of a system and their dependencies, illustrating the architecture and relationships between components.
- **Deployment Diagram:** Models the physical deployment of software components on hardware nodes, showing the distribution of system elements across computing resources.

2.3. UML Diagrams of Our System:

To accurately define and illustrate our chatbot system, we used two main UML diagrams: the Use Case Diagram and the Sequence Diagram. These diagrams elucidate the overall design, interconnections, and dynamic behavior of our program.

- **Uses Case Diagram**

- The use case diagram offers a comprehensive picture of the system's operation by delineating the primary players and their interactions with the system.
- Figure 3.1 illustrates that the key actor is the User, who interacts with the chatbot via a web interface. The user is able to upload documents (e.g., PDFs), pose inquiries, seek summaries, and extract certain information from the provided material.

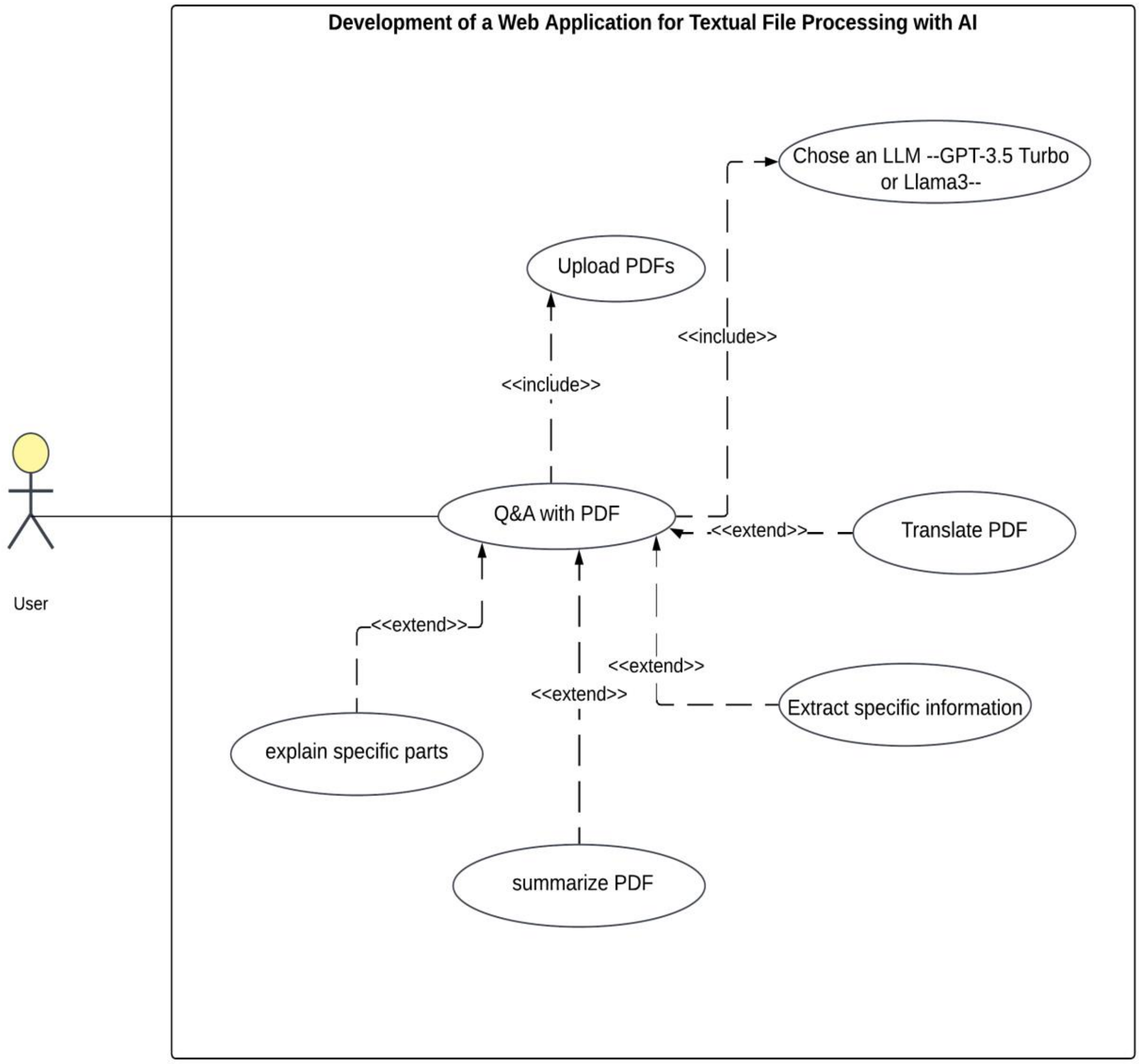


Figure III.1 Use Case Diagram for AI-Powered PDF Processing ChatBot

- **Sequence Diagram**

- Figure 3.2 presents a sequence diagram that delineates the interaction flow among the user, frontend, backend, embedding model, vector database (e.g., FAISS), and the language model API over a typical session.
- The user submits one or several papers using the chatbot interface.
- The backend processes the uploaded files by extracting text and dividing it into digestible segments.
- The segments are processed using an Embedding Model (e.g., all-MiniLM-L6-v2) to produce vector embeddings.
- The produced vectors are retained in an FAISS Vector Store to facilitate semantic search and retrieval.
- The backend informs the frontend that the system is prepared for inquiries.

- The individual poses an inquiry. The backend extracts the most relevant text segments from FAISS in accordance with the query.

- The chosen segments are provided as context to the Language Model API (e.g., GPT-3.5) to provide an appropriate answer.
- The Frontend presents the answer to the user.
- This graphic accurately illustrates the interactions and data flow required to address a user inquiry inside our chatbot system.

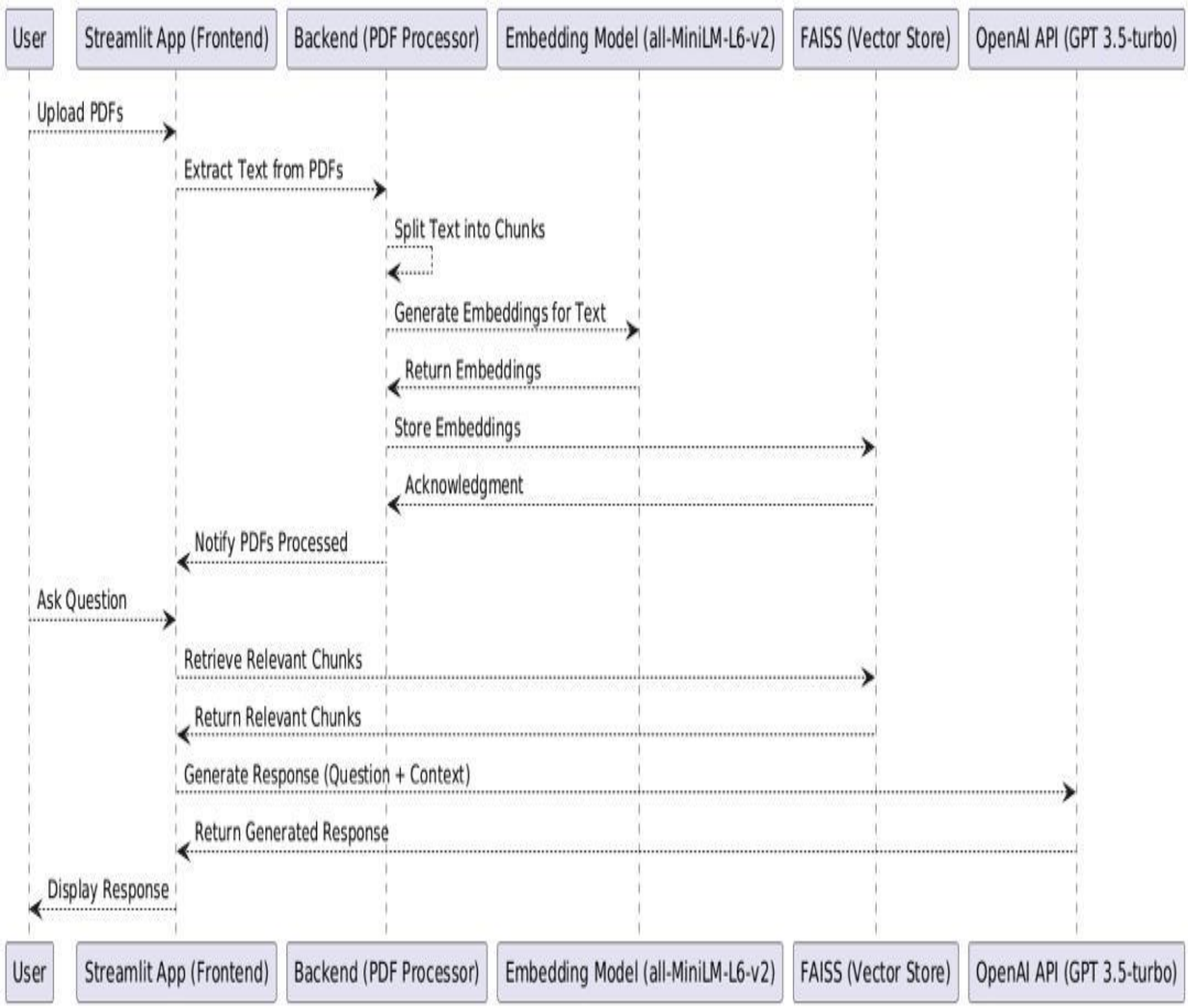


Figure III.2 Sequence Diagram of PDF Q&A System Workflow

3. Software and Tools Employed:

To implement the proposed system efficiently and ensure optimal performance across all stages—from document ingestion to intelligent response generation—a carefully selected set of software libraries, frameworks, and tools was utilized. Each tool plays a specific role in achieving modularity, scalability, and usability of the application.

3.1. Programming Language:

- **Python**: The entire system is developed using Python due to its extensive ecosystem of libraries in artificial intelligence, natural language processing, and web development. Python's simplicity and readability also facilitate rapid prototyping and maintenance [13].

Figure III.3 Python Programming Language Logo [56]

3.2. Front-End Framework:

- **Streamlit**: Used to build the user interface. Streamlit enables fast development of interactive web applications with minimal code and supports real-time updates, file upload widgets, and dynamic UI elements suitable for AI applications [29].

Figure III.4 Streamlit Framework Logo [56]

3.3. PDF Processing:

- **PyPDF2**: A Python library used to extract text content from uploaded PDF files. It supports reading, parsing, and navigating the structure of multi-page PDF documents, which is critical for the preprocessing phase [32].

Figure III.5 PyPDF2 PDF Processing Library [32]

3.4. Text Processing and Chunking:

- **LangChain**: A framework designed for developing applications powered by large language models. In this project, LangChain provides utilities such as “CharacterTextSplitter” for dividing documents into manageable text chunks and “ConversationalRetrievalChain” for managing conversational context and retrieval-based generation [25].

Figure III.6 LangChain Framework Architecture [25]

3.5. Semantic Embedding:

- **HuggingFace Transformers**: Specifically, the all-MiniLM-L6-v2 embedding model is used to convert textual chunks into high-dimensional vectors that capture their semantic meaning. These embeddings are essential for similarity search and context retrieval [33].

Figure III.7 Hugging Face Transformers Library [57]

3.6. Vector Database:

- **FAISS (Facebook AI Similarity Search)**: An open-source library developed by Facebook for efficient similarity search in large vector spaces. FAISS is used to index and search embedded document chunks, enabling fast and accurate retrieval of relevant information based on user queries [39].

3.7. Language Models (LLMs):

- **Llama 3 70B via Groq API:** A high-performance large language model developed by Meta, accessed through the Groq API for extremely fast inference and low-latency generation, making it suitable for real-time applications [12][34].

Figure III.8 Groq LPU Inference Engine Logo [58]

- **GPT-3.5-Turbo via OpenAI API:** An advanced model developed by OpenAI, known for its strong performance in understanding context and generating coherent responses. The API provides a reliable and scalable way to integrate the model into the system [35].

Figure III.9 OpenAI AI Research & API Logo [59]

3.8. Data Storage and Management:

- **JSON**: Used as an intermediate format for storing preprocessed text and metadata in a structured manner, enabling ease of access, transformation, and debugging during the pipeline execution [36].

3.9. Development and Deployment Tools:

- **VS Code**: A widely-used integrated development environment (IDE) that supports Python and offers extensions for Streamlit and LLM development [37].

Figure III.10 Visual Studio Code (VS Code) IDE Logo [60]

- **Virtual Environments (venv)**: Ensures dependency isolation and avoids conflicts between libraries during development [38].

4. Our Platform's Interface:

The user interface (UI) is a critical aspect of the proposed system, as it ensures ease of use, interactivity, and seamless user experience when interacting with the platform. The interface has been designed to provide a simple yet efficient way for users to upload PDF files, interact with the system, and obtain meaningful answers based on their queries. This section outlines the design, functionality, and features of the platform’s interface.

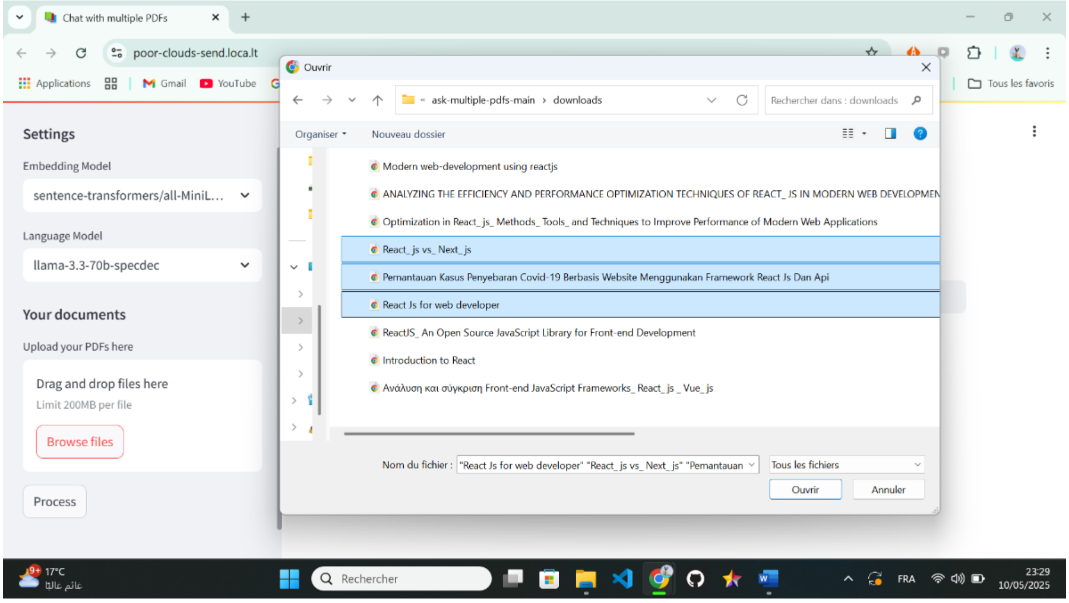
4.1. Design Principles:

The interface is built around a few key design principles:

- **Simplicity**: The UI is clean and intuitive, allowing users to quickly understand how to upload documents and interact with the system without requiring extensive technical knowledge.
- **Interactivity**: The system offers an interactive experience where users can query the system in natural language and receive responses in real-time.
- **Responsiveness**: The interface adapts to different screen sizes and devices, ensuring that it is accessible on both desktop and mobile platforms.

4.2. Main Features of the Interface:

- **File Upload** (figure III.11)**:** The first step in the user’s interaction is uploading PDF files. The platform allows the user to upload one or multiple PDF files at once. This is achieved through a simple drag-and-drop interface or by browsing their local filesystem. Once the files are uploaded, they are processed and stored for further analysis.



Chat with multiple PDFs

poor-clouds-send.loca.lt

Settings

Embedding Model: sentence-transformers/all-MiniL...

Language Model: llama-3.3-70b-specdec

Your documents

Upload your PDFs here

Drag and drop files here

Limit 200MB per file

Ouvrir

ask-multiple-pdfs-main › downloads

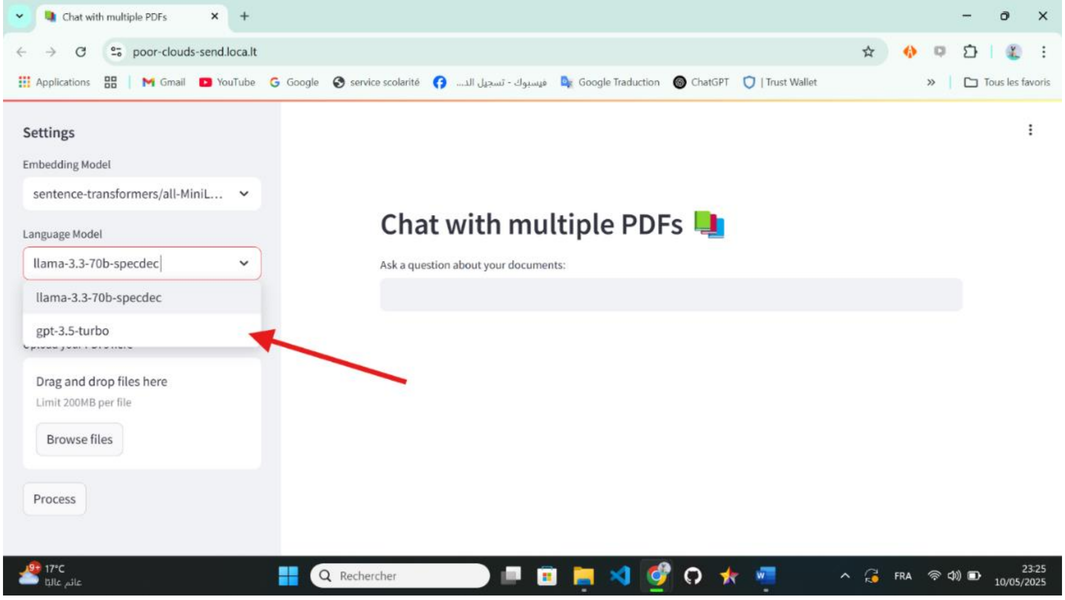
- Modern web-development using reactjs
- ANALYZING THE EFFICIENCY AND PERFORMANCE OPTIMIZATION TECHNIQUES OF REACT_ JS IN MODERN WEB DEVELOPMEN
- Optimization in React_js_ Methods_ Tools_ and Techniques to Improve Performance of Modern Web Applications
- React_js vs_ Next_js
- Pemantauan Kasus Penyebaran Covid-19 Berbasis Website Menggunakan Framework React Js Dan Api
- React Js for web developer
- ReactJS_ An Open Source JavaScript Library for Front-end Development
- Introduction to React
- Ανάλυση και σύγκριση Front-end JavaScript Frameworks_ React_js _ Vue_js

Nom du fichier : "React Js for web developer" "React_js vs_ Next_js" "Pemantauan

Tous les fichiers

Figure III.11 Multi-PDF Upload Interface

- **Model Selection** (figure III.12)**:** The platform allows users to select the language model they wish to use for generating responses. Currently, users can choose between:
    - Llama 3 70B for high-performance, real-time responses.
    - GPT-3.5Turbo for coherent and contextually relevant answers.
    This selection ensures that the user can tailor the system's behavior to their specific needs.



Chat with multiple PDFs

poor-clouds-send.loca.lt

Settings

Embedding Model: sentence-transformers/all-MiniL...

Language Model: llama-3.3-70b-specdec

- llama-3.3-70b-specdec
- gpt-3.5-turbo

Drag and drop files here

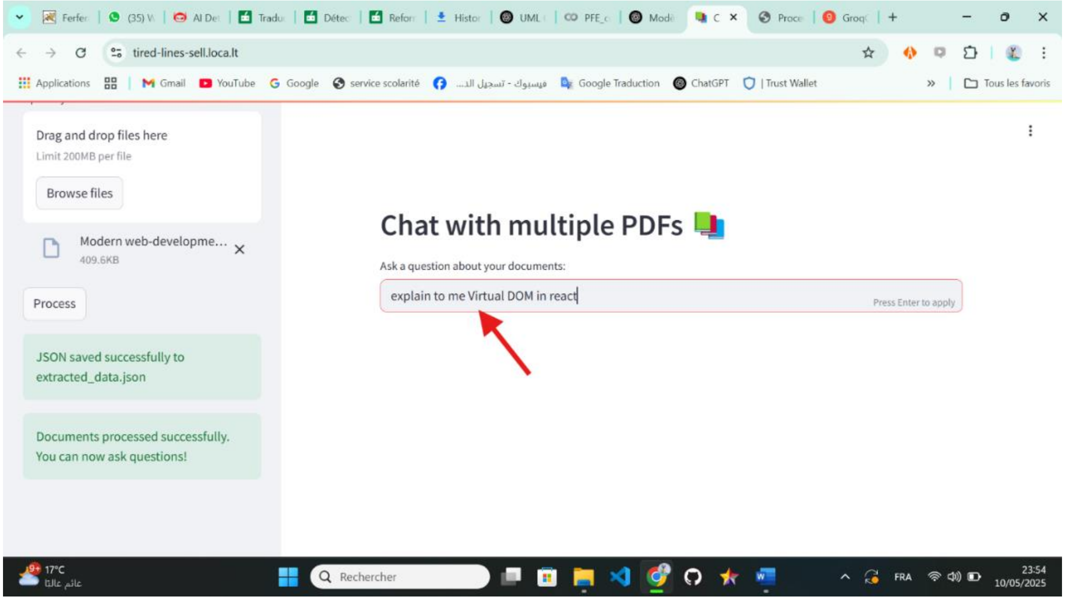
Limit 200MB per file

Chat with multiple PDFs

Ask a question about your documents:

Figure III.12 Model Selection and Multi-PDF Query Interface

- **Query Input** (figure III.13)**:** After uploading the PDF files, users can input their questions directly into a text box on the interface. The text input is designed to accept natural language queries, making it easy for users to ask questions about the contents of the PDFs without needing to understand the underlying technicalities.



JSON saved successfully to extracted_data.json

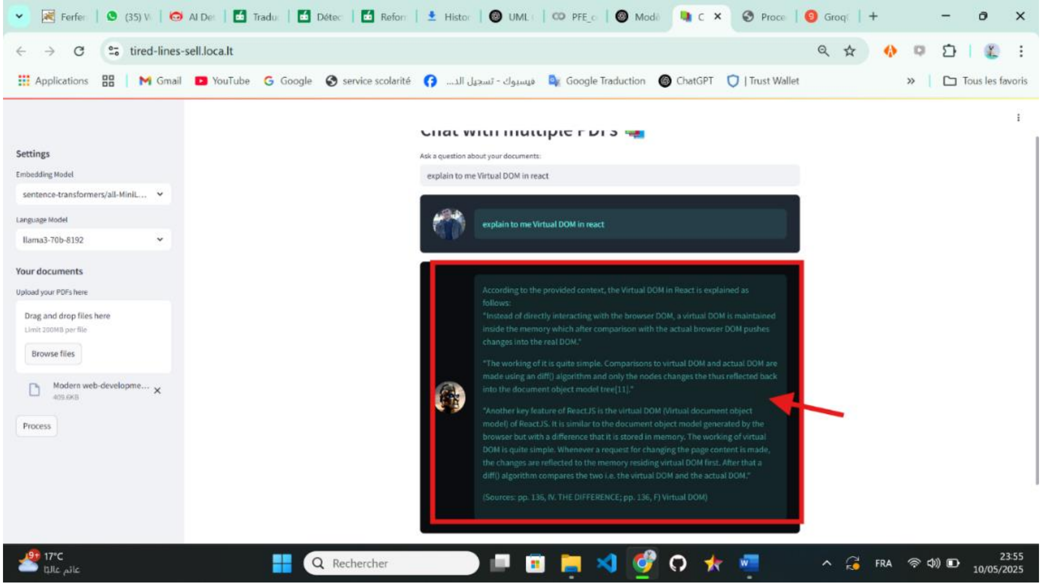
Documents processed successfully. You can now ask questions!

Ask a question about your documents:

explain to me Virtual DOM in react

Figure III.13 PDF Query Interface with Question Input

- **Dynamic Results Display** (figure III.14)**:** Once the user submits their query, the system processes the request, retrieves the most relevant information from the PDFs, and generates a response. The answers are displayed in a chat-like interface, with user inputs and model responses distinguished by different colors and backgrounds. This allows for an easy-to-follow, dynamic conversation between the user and the system. Each response is accompanied by its source, enabling users to trace the information back to the original document for verification and deeper understanding.



Embedding Model: sentence-transformers/all-MiniL…

Language Model: llama3-70b-8192

Ask a question about your documents:

explain to me Virtual DOM in react

explain to me Virtual DOM in react

According to the provided context, the Virtual DOM in React is explained as follows:
"Instead of directly interacting with the browser DOM, a virtual DOM is maintained inside the memory which after comparison with the actual browser DOM pushes changes into the real DOM."

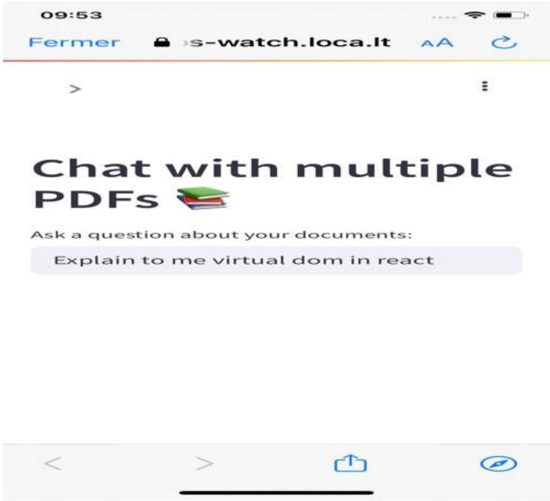
"The working of it is quite simple. Comparisons to virtual DOM and actual DOM are made using an diff() algorithm and only the nodes changes the thus reflected back into the document object model tree[11]."

"Another key feature of ReactJS is the virtual DOM (Virtual document object model) of ReactJS. It is similar to the document object model generated by the browser but with a difference that it is stored in memory. The working of virtual DOM is quite simple. Whenever a request for changing the page content is made, the changes are reflected to the memory residing virtual DOM first. After that a diff() algorithm compares the two i.e. the virtual DOM and the actual DOM."

(Sources: pp. 136, IV. THE DIFFERENCE; pp. 136, F) Virtual DOM)

Figure III.14 Live User Activity Display

- **Mobile Responsiveness** (figure III.15)**:** The platform is fully responsive and optimized for use on mobile devices. Whether accessed via a smartphone or tablet, the interface adapts seamlessly to different screen sizes, ensuring a smooth and intuitive user experience. Features like file upload, model selection, and query input remain fully functional, allowing users to interact with the system conveniently on the go.



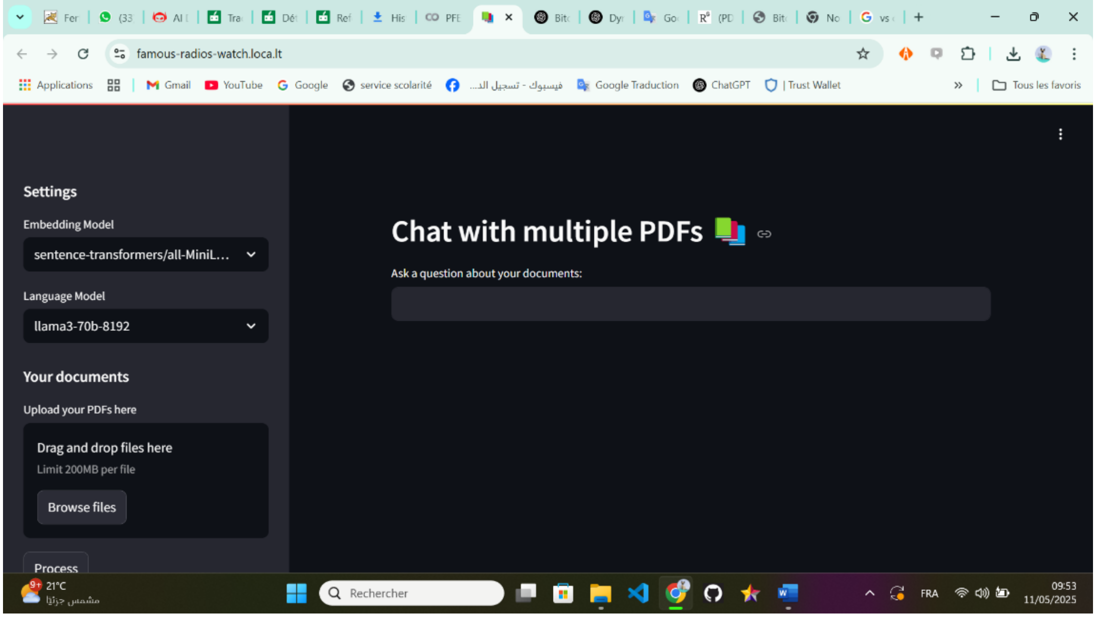
Chat with multiple PDFs

Ask a question about your documents:

Explain to me virtual dom in react

Figure III.15 Mobile-Responsive Activity Dashboard

- **Dark Mode Support** (figure III.16)**:** The platform includes a dark mode option, providing users with a visually comfortable experience in low-light environments. Users can easily switch between light and dark themes according to their preferences, reducing eye strain and enhancing readability during extended use.



Settings

Embedding Model

sentence-transformers/all-MiniL...

Language Model

llama3-70b-8192

Your documents

Upload your PDFs here

Drag and drop files here

Limit 200MB per file

Browse files

Process

Chat with multiple PDFs

Ask a question about your documents:

Figure III.16 Dark Mode Interface

5. Experiments and Results:

5.1. Experimental Setup:

To evaluate the capability of our AI-based document understanding system, we conducted a set of experiments using two different text files:

1. A personal CV of a student at Tahri Mohamed University of Béchar.
2. The book "Rich Dad Poor Dad" by Robert T. Kiyosaki.

Both files were used to test how well the model could extract information, summarize content, and answer user queries accurately.

- Model Used: Llama 3 via Groq API
- Frameworks: LangChain + FAISS + HuggingFace Embeddings
- Languages: English

5.2. Processing Pipeline:

Each file was handled through the following steps:

1. Text Ingestion: The content of both files was extracted and preprocessed using LangChain’s TextLoader.
2. Text Splitting: The documents were split into chunks of ~500 tokens with overlap.
3. Embedding: Text chunks were embedded using HuggingFace’s BGE or All-MiniLM models.
4. Vector Storage: Embeddings were stored in a FAISS vector store for semantic retrieval.
5. Query Handling: A user input question was matched with the most relevant chunks and passed to Llama 3 for response generation.

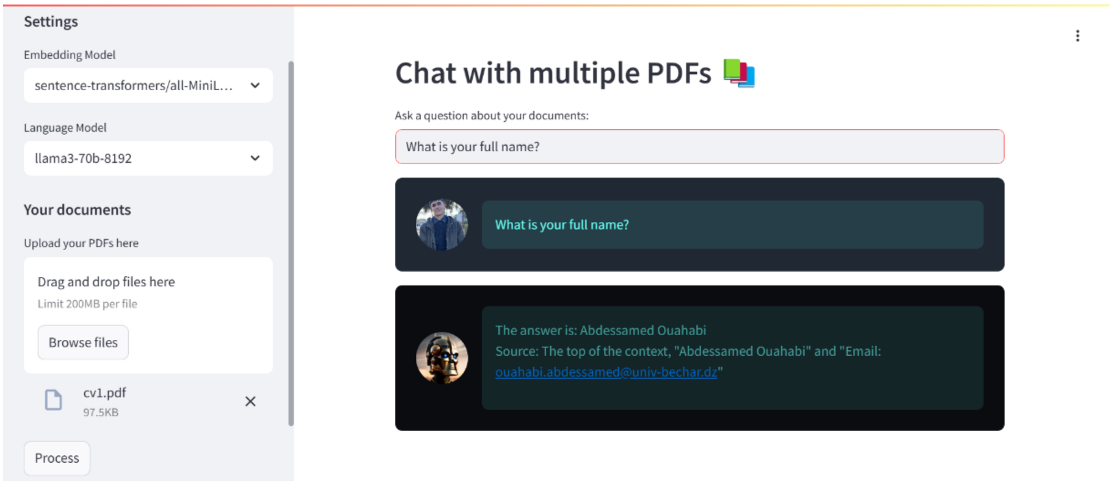
5.3. Test Cases (Simplified):

We tested the system using two different types of documents:

5.3.1. CV File (Abdessamed Ouahabi – Student at University of Béchar):

We asked the model simple questions such as (figure III.17; III.18 ;III.19):

- **Question:** What is your full name?
  Answer: The answer is: Abdessamed Ouahabi



Embedding Model: sentence-transformers/all-MiniL…

Language Model: llama3-70b-8192

Your documents

Upload your PDFs here

Drag and drop files here
Limit 200MB per file

cv1.pdf 97.5KB

Chat with multiple PDFs 📚

Ask a question about your documents:

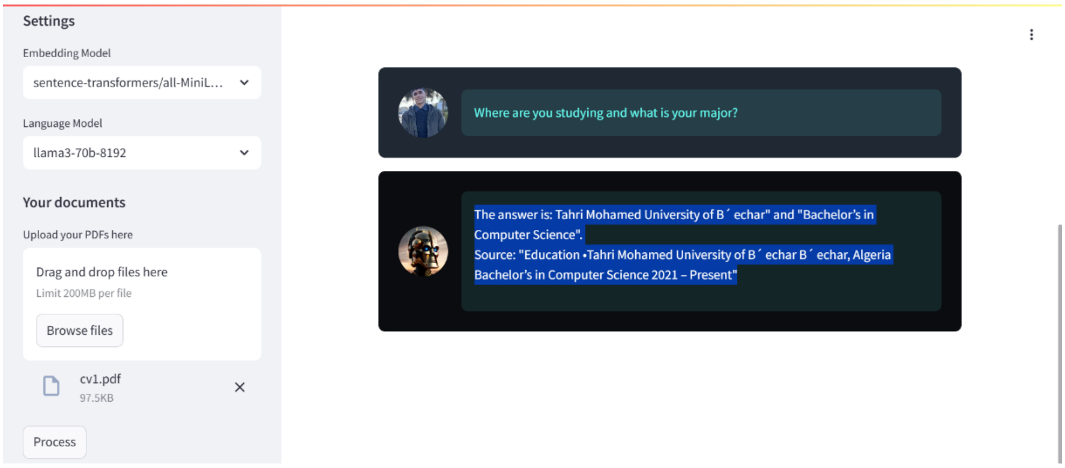
What is your full name?

What is your full name?

The answer is: Abdessamed Ouahabi
Source: The top of the context, "Abdessamed Ouahabi" and "Email: ouahabi.abdessamed@univ-bechar.dz"

Figure III.17 Response Generated by chatbot 01

- **Question:** Where are you studying and what is your major?
  Answer: The answer is: Tahri Mohamed University of Béchar and "Bachelor's in Computer Science".



Embedding Model: sentence-transformers/all-MiniL…

Language Model: llama3-70b-8192

Your documents

Upload your PDFs here

Drag and drop files here
Limit 200MB per file

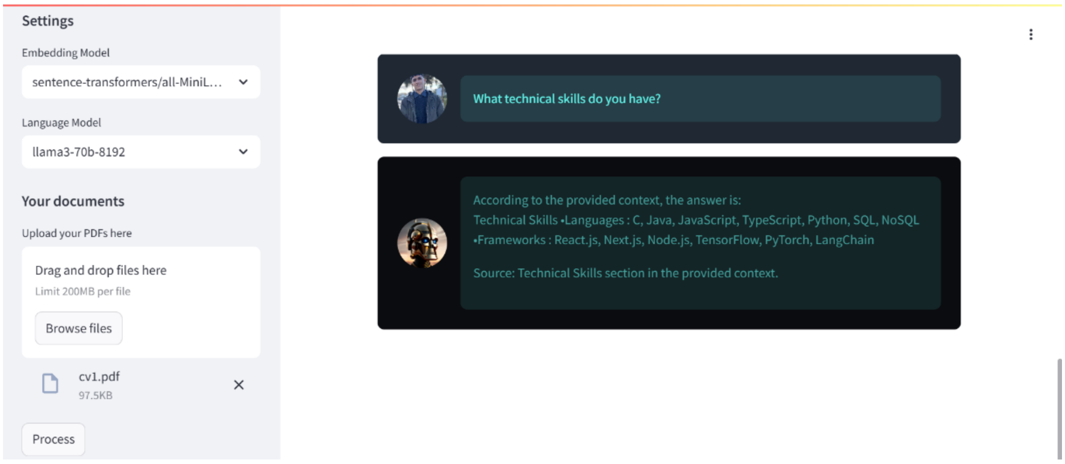
cv1.pdf 97.5KB

Where are you studying and what is your major?

The answer is: Tahri Mohamed University of B´echar" and "Bachelor's in Computer Science".
Source: "Education •Tahri Mohamed University of B´echar B´echar, Algeria Bachelor's in Computer Science 2021 – Present"

Figure III.18 Response Generated by chatbot 02

- **Question:** What technical skills do you have?
  Answer:
  Technical Skills
  +Languages: C, Java, JavaScript, TypeScript, Python, SQL, NoSQL
  +Frameworks: React.js, Next.js, Node.js, TensorFlow, PyTorch, LangChain
  Source: Technical Skills section in the provided context.



Settings

Embedding Model: sentence-transformers/all-MiniL…

Language Model: llama3-70b-8192

Your documents

Upload your PDFs here

cv1.pdf 97.5KB

What technical skills do you have?

According to the provided context, the answer is:
Technical Skills •Languages : C, Java, JavaScript, TypeScript, Python, SQL, NoSQL •Frameworks : React.js, Next.js, Node.js, TensorFlow, PyTorch, LangChain

Source: Technical Skills section in the provided context.

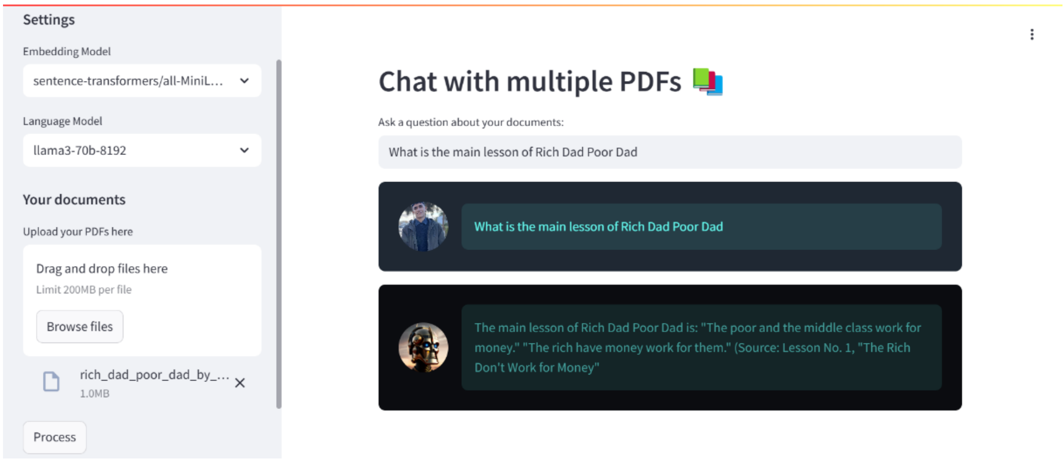
Figure III.19 Response Generated by chatbot 03

The model was able to extract all the key information from the CV correctly and quickly.

5.3.2. Book: "Rich Dad Poor Dad":

We then tested the model with a full book to see how it handles larger, unstructured content. Example questions (figure III.20; III.21; III.22):

1. Question: What is the main lesson of Rich Dad Poor Dad?
   Answer: The main lesson is: "The poor and the middle class work for money. The rich have money work for them.



Settings

Embedding Model: sentence-transformers/all-MiniL…

Language Model: llama3-70b-8192

Your documents

Upload your PDFs here

rich_dad_poor_dad_by_… 1.0MB

Chat with multiple PDFs

Ask a question about your documents: What is the main lesson of Rich Dad Poor Dad

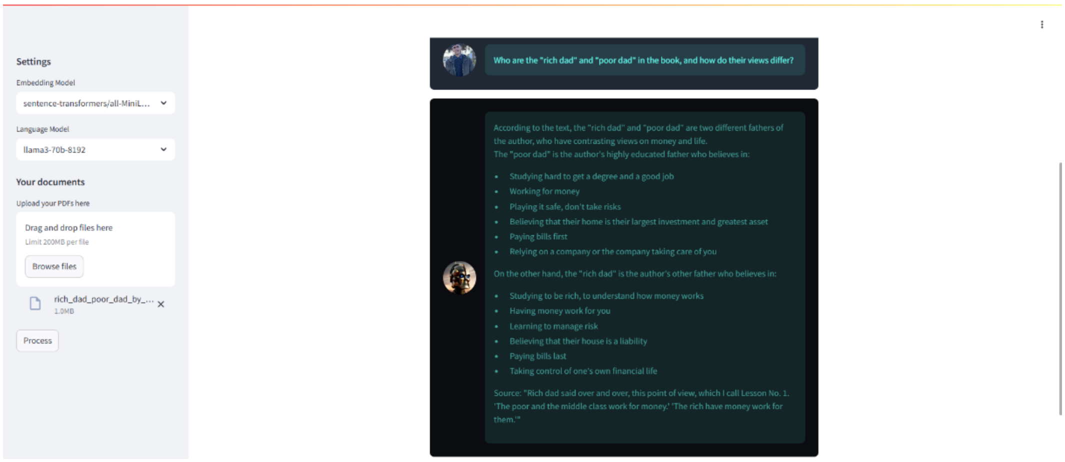
What is the main lesson of Rich Dad Poor Dad

The main lesson of Rich Dad Poor Dad is: "The poor and the middle class work for money." "The rich have money work for them." (Source: Lesson No. 1, "The Rich Don't Work for Money"

Figure III.20 Response Generated by chatbot 04

2. Question: Who are the "rich dad" and "poor dad" in the book, and how do their views differ?
   Answer:
   - Poor Dad: The author's biological father, who believes in:
     - Studying hard to get a degree and a secure job.
     - Working for money and playing it safe.

    - Viewing a home as the largest investment.
    - Paying bills first and relying on employers for security.
  - Rich Dad: The author's mentor, who believes in:
    - Studying to understand how money works.
    - Making money work for you and managing risk.
    - Viewing a home as a liability.
    - Paying bills last and taking control of one's financial life. Source: "Rich dad said... The poor and the middle class work for money. The rich have money work for them."



Settings

Embedding Model: sentence-transformers/all-MiniL...

Language Model: llama3-70b-8192

Your documents

Upload your PDFs here

Who are the "rich dad" and "poor dad" in the book, and how do their views differ?

According to the text, the "rich dad" and "poor dad" are two different fathers of the author, who have contrasting views on money and life.
The "poor dad" is the author's highly educated father who believes in:

- Studying hard to get a degree and a good job
- Working for money
- Playing it safe, don't take risks
- Believing that their home is their largest investment and greatest asset
- Paying bills first
- Relying on a company or the company taking care of you

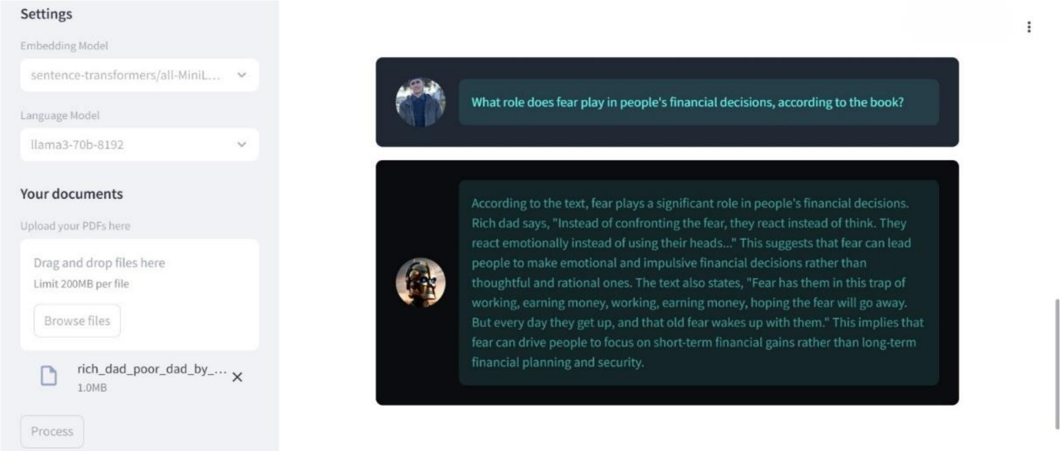
On the other hand, the "rich dad" is the author's other father who believes in:

- Studying to be rich, to understand how money works
- Having money work for you
- Learning to manage risk
- Believing that their house is a liability
- Paying bills last
- Taking control of one's own financial life

Source: "Rich dad said over and over, this point of view, which I call Lesson No. 1. 'The poor and the middle class work for money.' 'The rich have money work for them.'"

Figure III.21 Response Generated by chatbot 05

3. Question: What role does fear play in people’s financial decisions, according to the book?
   Answer:
   Fear drives people to make emotional, impulsive decisions rather than rational ones. It traps them in a cycle of working for short-term security, hoping fear will disappear, but it persists.
   Key Quotes:
   - "Fear has them in this trap of working, earning money, working, earning money..."
   - "They react emotionally instead of using their heads."
     Source: Excerpts discussing fear and financial behavior.



Settings

Embedding Model: sentence-transformers/all-MiniL...

Language Model: llama3-70b-8192

Your documents

Upload your PDFs here

rich_dad_poor_dad_by_... 1.0MB

What role does fear play in people's financial decisions, according to the book?

According to the text, fear plays a significant role in people's financial decisions. Rich dad says, "Instead of confronting the fear, they react instead of think. They react emotionally instead of using their heads..." This suggests that fear can lead people to make emotional and impulsive financial decisions rather than thoughtful and rational ones. The text also states, "Fear has them in this trap of working, earning money, working, earning money, hoping the fear will go away. But every day they get up, and that old fear wakes up with them." This implies that fear can drive people to focus on short-term financial gains rather than long-term financial planning and security.

Figure III.22 Response Generated by chatbot 06

5.4. Key Observations from Our Trials:

- **Effective handling of structured documents:**
  When tested with a structured file like a CV, the system was able to recognize and extract information with high accuracy.
- **Accurate responses to specific queries:**
  The model correctly answered questions related to the student's major, technical skills, and completed projects demonstrating consistent and reliable performance.
- **Understanding of document structure:**
  The system showed that it understands structured formats and can retrieve answers from specific sections of a document.
- **Successful interaction with long, unstructured text:**
  When tested with the full book *Rich Dad Poor Dad*, the model also performed well, despite the complexity and length of the content.
- **Conceptual understanding of the book:**
  The model was able to:
  - Summarize chapters
  - Explain financial concepts (e.g., assets vs. liabilities)
  - Compare the views of “rich dad” and “poor dad” clearly
- **Fast response time:**
  In all test cases, responses were generated in under two seconds, which is suitable for real-time or interactive use.
- **High linguistic quality:**
  The language used in the responses was clear, natural, and understandable, even for non-technical users.
- **Capability across different document types:**
  The model demonstrated strong performance on both short structured documents and long complex texts.
- **Practical applications:**
  These findings suggest that such AI systems can be valuable tools for students, researchers, and anyone needing quick and intelligent interaction with document content.

6. Conclusion:

In this chapter, we presented the technical foundation and implementation details of our proposed system. We began by outlining the system architecture through diagrams and design models, which clarified the interaction between the main components and their roles. Then, we detailed the software tools and frameworks that were selected to build and support the platform, highlighting their relevance and efficiency.

We also provided an overview of the platform's interface, showcasing its usability and how users interact with the document processing system. Furthermore, through experimental evaluations, we demonstrated the model’s ability to handle both structured and unstructured documents, delivering accurate and timely responses to user queries.

Overall, the chapter confirms that the proposed approach is technically feasible, effectively designed, and capable of providing intelligent document interaction in real-world scenarios.

General conclusion

General Conclusion:

The advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP) has significantly transformed how humans interact with digital information. One of the pressing challenges in this digital era is navigating and extracting relevant information from large and often unstructured textual data formats, such as PDF documents. Traditional search methods relying on keyword matching fall short in understanding context, leading to inefficient and sometimes frustrating user experiences.

In response to this problem, our project aimed to develop a web based AI chatbot capable of processing and interacting with PDF files using natural language queries. By leveraging frameworks such as Streamlit and LangChain, and integrating powerful Large Language Models (LLMs) like GPT-3.5 Turbo and Llama 3, we created a system that not only retrieves answers accurately but also summarizes content and explains complex segments clearly. This demonstrates how conversational AI can serve as an effective assistant in document analysis and comprehension.

The system architecture was carefully designed with modular components to handle file ingestion, semantic text embedding, query processing, and dynamic answer generation. Technologies such as FAISS for vector search and HuggingFace models for embeddings played essential roles in ensuring performance and scalability. Additionally, user experience was prioritized through an intuitive interface that allows seamless file upload and conversational interaction.

Throughout this project, we explored various AI concepts, from neural networks and transformer-based models to NLP preprocessing and document-aware architectures. This multidisciplinary integration highlights the importance of aligning theoretical understanding with practical development in order to build impactful applications.

In conclusion, this work not only provides a valuable tool for document exploration but also opens the door for further research in areas such as multimodal input (text, image, table), offline deployment, multilingual capabilities, and real-time applications. It serves as a concrete example of how AI, when thoughtfully implemented, can simplify complex tasks and enhance access to knowledge.

Bibliography:

[1]: Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning". Cambridge, MA: MIT Press, 2016. Available: https://www.deeplearningbook.org/ , (Consulted February 11, 2025)

[2]: Daniel Jurafsky and James H. Martin. 2025. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models", 3rd edition. Online manuscript released January 12, 2025. https://web.stanford.edu/~jurafsky/slp3. (Consulted February 11, 2025)

[3]: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser and Illia Polosukhin, "Attention is All you Need" in Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc. Eds: I. Guyon and U. Von Luxburg and S. Bengio and H. Wallach and R. Fergus and S. Vishwanathan and R. Garnett,(vol 30), 2017, (pp. 5998–6008)

[4]: Tom M. Mitchell, "Machine Learning", March 1, 1997

[5]: Yann LeCun, Yoshua Bengio and Geoffrey Hinton, "Deep learning" Nature, May 2015, (pp. 436–444), Available: https://www.researchgate.net/publication/277411157_Deep_Learning , (Consulted February 13, 2025)

[6]: Cole Stryker and Jim Holdsworth, "What is NLP (natural language processing)?", IBM, https://www.ibm.com/think/topics/natural-language-processing, (Consulted February 15, 2025)

[7]: Cole Stryker and Eda Kavlakoglu, "What is artificial intelligence (AI)?", IBM, https://www.ibm.com/think/topics/artificial-intelligence , (Consulted February 15, 2025)

[8]: Fran Škavić, "The implementation of artificial intelligence and its future potential", Undergraduate thesis, University of Zagreb, Zagreb, 2019

[9]: Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach", 3rd Edition, Upper Saddle River, New Jersey: Pearson, 2010

[10]: Margaret A. Boden, "Artificial Intelligence: A Very Short Introduction", Oxford University Press, 2018

[11]: Nils J. Nilsson, "THE QUEST FOR ARTIFICIAL INTELLIGENCE A HISTORY OF IDEAS AND ACHIEVEMENTS", Cambridge University Press, October 2009

[12]: "Llama 3 models", Meta, https://www.llama.com/models/llama-3/ , (Consulted May 03, 2025)

[13]: BENMOUFFOK SAID Oualid and LOUNAS Mohamed Sofiene, "Creation of a Chatbot for video games recommendation with NLP", Master thesis, University of Blida 1 – SAAD DAHLAB, Blida, 2020

[14]: Samiha Attaoua and Chaima Guesmia, "A multilingual chatbot for supporting mobile companies complaints. Case study: ATM Mobilis of Algeria.", Master thesis, UNIVERSITY MOHAMED BOUDIAF - M'SILA, M'sila, 2022

[15]: Nadjmeddine BOUDJELLAL, "Design and Implementation of a Customer service chatbot using deep learning approach", Master thesis, UNIVERSITY MOHAMED BOUDIAF - M'SILA, M'sila, 2018

[16]: KHICHA YAHIA and HARIZ MOFDI, “Proposal of a model for an intelligent behavior of a CHATBOT”, Master thesis, UNIVERSITY OF ECHAHID HAMMA LAKHDAR - EL OUED, El oued, 2021

[17]: Soham Roy, Mitul Goswami, Nisharg Nargund, Suneeta Mohanty and Prasant Kumar Pattnaik, “Conversational Text Extraction with Large Language Models Using Retrieval-Augmented Systems”, arXiv preprint, arXiv:2501.09801, Jan 2025

[18]: Malay Agarwal, “Pre-training Large Language Models”, available: https://raw.githubusercontent.com/MalayAgr/generative-ai-with-llms-notes/828942e16110211fdf9e8a0d54ec291a54a5a211/Week%201/Pre-training%20Large%20Language%20Models.pdf , (Consulted February 24, 2025)

[19]: Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai and Xuanjing Huang, “Pre-trained Models for Natural Language Processing: A Survey”, arXiv preprint, arXiv: 2003.08271v4, 23 Jun 2021

[20]: Marie Gobiet, “The History Of Chatbots – From ELIZA to ChatGPT”, Onlim, https://onlim.com/en/the-history-of-chatbots/ , (Consulted May 08, 2025)

[21]: “Over 11,000 chatbots created for Facebook’s Messenger platform”, Ciol, https://www.ciol.com/over-11000-chatbots-created-for-facebooks-messenger-platform/ , (Consulted May 08, 2025)

[22]: “A Brief History of LLMs From Transformers (2017) to DeepSeek-R1 (2025)”, Medium, https://medium.com/@lmpo/a-brief-history-of-lmms-from-transformers-2017-to-deepseek-r1-2025-dae75dd3f59a , (Consulted May 08, 2025)

[23]: Ross W. Green, “This history of Retrieval-Augmented Generation in 3 minutes…!”, Medium, https://medium.com/@custom_aistudio/this-history-of-retrieval-augmented-generation-in-3-minutes-f7f07073599a , May 08, 2025

[24]: “GPT-4 Turbo”, OpenAI, https://platform.openai.com/docs/models/gpt-4-turbo , (Consulted May 09, 2025)

[25]: “Introduction”, LangChain, https://python.langchain.com/docs/introduction/ , (Consulted April 21, 2025)

[26]: Jeffrey Erickson, “The Role and Benefits of AI in Cloud Computing”, Oracle, https://www.oracle.com/artificial-intelligence/ai-cloud-computing/#explained , (Consulted May 12, 2025)

[27]: “What you can build with Rasa”, Rasa, https://rasa.com/docs/learn/use-cases , (Consulted May 12, 2025)

[28]: “Introduction to Rasa X/Enterprise”, Rasa, https://legacy-docs-enterprise.rasa.com/docs/rasa-enterprise/ , (Consulted May 12, 2025)

[29]: “A faster way to build and share data apps”, streamlit, https://streamlit.io/ , (Consulted May 12, 2025)

[30]: “What’s New In Python 3.13”, Python, https://docs.python.org/3/whatsnew/3.13.html , (Consulted May 12, 2025)

[31]: Sunguk Lee, "Unified Modeling Language (UML) for database systems and computer applications," Int. J. Database Theory Appl., vol. 5, no. 1, pp. 157–164, 2012.

[32]: "Welcome to PyPDF2" PyPDF2, https://pypdf2.readthedocs.io/en/3.x/ (Consulted May 11, 2025).

[33]: "Transformers", Hugging Face, https://huggingface.co/docs/transformers/index , (Consulted May 11, 2025)

[34]: "The Shift to AI Inference", groq, https://groq.com/about-us/ , (Consulted May 16, 2025).

[35]: "GPT-3.5 Turbo" OpenAI, https://platform.openai.com/docs/models/gpt-3.5-turbo , (Consulted May 11, 2025).

[36]: "Introducing JSON,", JSON, https://www.json.org (Consulted May 11, 2025).

[37]: "Visual Studio Code documentation,", Visual Studio Code, https://code.visualstudio.com/docs (Consulted May 11, 2025).

[38]: "venv — Creation of virtual environments", Python, https://docs.python.org/3/library/venv.html , (Consulted May 11, 2025).

[39]: “Faiss”, Meta, https://ai.meta.com/tools/faiss/ , (Consulted May 16, 2025).

[40]: Harry Guinness, “What is artificial general intelligence (AGI)?”, Zapier, https://zapier.com/blog/artificial-general-intelligence/ , (Consulted April 11, 2025)

[41]: “Reactive Machines”, BotPenguin, https://botpenguin.com/glossary/reactive-machines , (Consulted April 11, 2025)

[42]: Nisha Arya, “Three Types of Machine Learning”, Ejable, https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/types-of-machine-learning/ , (Consulted April 11, 2025)

[43]:Figure I.6 “Le "Machine Learning" est-il semblable à l’IA?”, Google News Initiative, https://newsinitiative.withgoogle.com/fr-fr/resources/trainings/is-machine-learning-the-same-thing-as-ai/ , (Consulted April 11, 2025)

[44]: Tavish Srivastava, “How does Artificial Neural Network (ANN) algorithm work? Simplified!”, Analytics Vidhya, https://www.analyticsvidhya.com/blog/2014/10/ann-work-simplified/ , (Consulted April 11, 2025)

[45]: “Appréhendez le Deep Learning ou l'apprentissage profond”, OPENCLASSROOMS, https://openclassrooms.com/fr/courses/6417031-objectif-ia-initiez-vous-a-lintelligence-artificielle/6823506-apprehendez-le-deep-learning-ou-lapprentissage-profond , (Consulted April 11, 2025)

[46]: “Impact of Data Science in Robotics”, USDSI, https://www.usdsi.org/data-science-insights/impact-of-data-science-in-robotics , (Consulted March 16, 2025)

[47]: “Expert Systems in AI – Components, Applications and Characteristics”, TechVidvan, https://techvidvan.com/tutorials/expert-systems-in-ai/ , (Consulted April 11, 2025)

[48]: “14 Natural Language Processing Techniques Evolving the NLP Industry”, Revolveai, https://revolveai.com/natural-language-processing-techniques/ , (Consulted April 11, 2025)

[49]: Abirami Vina, “Du code à la conversation : Comment fonctionne un LLM ?”, ultralytics, https://www.ultralytics.com/fr/blog/from-code-to-conversation-how-does-an-llm-work , (Consulted April 11, 2025)

[50]: Arnavgupta, “Tokenization and it’s application”, Medium, https://medium.com/@arnavgupta16092004/tokenization-and-its-application-69ce6d90ed13 , (Consulted April 22, 2025)

[51]: “Stemming and Lemmatization”, Study Machine Learning, https://studymachinelearning.com/stemming-and-lemmatization/ , (Consulted April 22, 2025)

[52]: Matthew Kramer, “Parts of Speech (POS) Tagging”, Medium, https://medium.com/codex/parts-of-speech-pos-tagging-98f18693ebc0 , (Consulted April 22, 2025)

[53]: Lucia Maria Coppola, “Learning about named entity recognition use cases”, DATAVID, https://datavid.com/blog/named-entity-recognition-use-cases , (Consulted April 22, 2025)

[54]: “Removing stop words with NLTK in Python”, geeksforgeeks, https://www.geeksforgeeks.org/removing-stop-words-nltk-python/ , (Consulted April 22, 2025)

[55]: “Introduction to Seq2Seq Models”, Analytics Vidhya, https://www.analyticsvidhya.com/blog/2020/08/a-simple-introduction-to-sequence-to-sequence-models/ , (Consulted April 22, 2025)

[56]: “Streamlit Deployments Unleashed”, Medium, https://medium.com/@mohankrishce/streamlit-deployments-unleashed-f5078d145462 , (Consulted may 16, 2025)

[57]: “Download Hugging Face logo transparent PNG”, StickPNG, https://www.stickpng.com/img/icons-logos-emojis/tech-companies/hugging-face-logo , (Consulted may 16, 2025)

[58]: “AI Chip Startup Groq Reaches $2.8B Valuatio”, Open Data Science, https://opendatascience.com/ai-chip-startup-groq-reaches-2-8b-valuation/ , (Consulted may 16, 2025)

[59]: “File:OpenAI Logo.svg”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:OpenAI_Logo.svg , (Consulted may 16, 2025)

[60]: “File:Visual Studio Code 1.35 icon.svg”, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Visual_Studio_Code_1.35_icon.svg , (Consulted may 16, 2025)