INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Inteligência Artificial na Comunicação Corporativa: modelos avançados para otimizar em tempo real interações

MARIANA PATRÍCIA NEVES MIRANDA
Setembro de 2024

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Artificial Intelligence in Corporate Communication: advanced models to optimize real-time interactions

Mariana Patrícia Neves Miranda

Dissertation to obtain the master's degree in computer engineering, area of specialization in cybersecurity.

Orientador: Isabel Praça

Co-orientador: Vladimiro Ferreira

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, janeiro 2023

Integrity statement

I declare that I have conducted this academic work with integrity.

I have not plagiarized or applied any form of misuse of information or falsification of results throughout the process that led to its elaboration.

Therefore, the work presented in this document is original and my own and has not previously been used for any other purpose.

I further declare that they are fully aware of the code of ethical conduct of P. PORTO.

ISEP, Porto, 15 de setembro de 2024

Dedicatory

I dedicate this thesis to all those who, directly or indirectly, stood by my side and contributed to making this moment a reality.

To my parents, who not only gave me the opportunity to dream but also provided the support and means to turn those dreams into reality. To you, I express my gratitude for every piece of advice, every word of encouragement, every sacrifice, and for teaching me the true meaning of dedication and perseverance. Without you, none of this would have been possible. To my brother, whose friendship and companionship brought me balance in challenging moments and joy in times of achievement.

To my boyfriend, for being my greatest support, for believing in me when I doubted myself, and for walking by my side with patience, love, and understanding. Your faith in me and your encouragement were crucial in helping me move forward, even when the challenges seemed overwhelming.

To all of you, I dedicate this achievement with love, gratitude, and deep appreciation for everything you have meant to me during this journey.

Resumo

As empresas são compostas por diversos colaboradores e os processos de suporte são fundamentais para o bom funcionamento da empresa. Infelizmente, surgem desafios quando a disponibilidade da equipa de suporte é reduzida, resultando em tempos de espera prolongados para os funcionários resolverem os seus problemas.

Em resposta a este desafio, será concebido um agente conversacional proativo para responder às perguntas mais frequentes dos funcionários. Esta solução inovadora visa agilizar o processo de suporte e melhorar a experiência do utilizador.

O agente consistirá em um módulo de interpretação textual, utilizando métodos de inteligência artificial como aprendizagem automática, processamento de linguagem natural e *Large Language Model*. Este agente garante uma interação mais ágil e precisa com os funcionários, reduzindo, em última análise, a dependência de recursos de suporte humano e acelerando a resolução de problemas para os mesmos.

Ao implementar este agente conversacional inteligente, pretende-se otimizar os serviços de suporte, mitigar o tempo de espera e melhorar a eficiência operacional geral.

Palavras-chave: Inteligência Artificia, Large Language Model, Machine learning, Processamento de Linguagem Natural.

Abstract

Support processes are crucial for the smooth operation of a company. Unfortunately, challenges arise when the availability of the support team is reduced, resulting in prolonged wait times for employees to resolve their issues.

In response to this challenge, a proactive conversational agent will be designed to answer the most frequently asked questions by the employees. This innovative solution aims to streamline the support process and enhance user experience.

The agent will consist of a text language interpretation module, using artificial intelligence methods, such as machine learning, natural language processing and Large Language Models. This ensures a more responsive and accurate interaction with users, ultimately reducing the dependency on human support resources and speed up issue resolution for employees.

By implementing this intelligent conversational agent, we aim to optimize support services, mitigate waiting time, and enhance overall operational efficiency.

Keywords: Intelligence Artificial, Large Language Model, Machine Learning, Natural Language Processing.

Index

List of Figures

List of tables

Acronyms and Symbols

List of Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BiLM	Bidirectional language model
CNN	Convolution Neural Network
GPT	Generative pre-trained transformer
IT	Information Technology
LLM	Large Language Model
LSTMs	Long-Short-Term memory units
ML	Machine Learning
NLP	Natural Language Processing
RAG	Retrieval Augmented Generation
RNN	Recurrent Neural Networks

1 Introduction

This chapter aims to give an overview of the project, so it begins by explaining the context, explains the problem, the main objectives, and the document structure.

1.1 Context

In an era where technology is in a constant state of evolution and the increasing complexity of demands in the corporate environment highlight the urgent need to improve the technical support processes.

With the expansion of virtual services, new methods of end-user support have emerged, particularly the use of intelligent agents for clarifying doubts. Essentially, these agents aim to simulate a human assistant and, through artificial intelligence, comprehend and respond appropriately to user's inquiries [1]. Typically, they can get involve in continuous dialogues, but they can also be employed through visual or auditory communication.

Its popularity is noteworthy, not only due to the shift from the physical to the virtual system, but also because it brings advantages over traditional user's support methods [2]. There are several points of advantages in implementing an intelligent solution compared to the current system in place, like: associated costs, the liberation of the support team for other tasks, the operational efficiency, quick response, and the standardization of information.

The project for this dissertation was developed in SEG Automotive Portugal, Lda, for the Information Technology (IT) department. The main purposes of this project are to develop an innovative, efficient, and intelligent agent to automate and help the helpdesk team responsible for supporting the employees. This agent focusses on the integration of advanced technologies such as Artificial Intelligence (AI), Natural Language Processing (NLP), and Large Language Models (LLMs). This agent will interact with the users, aiding and supporting the helpdesk team and the employees of the company.

1.1.1 SEG Automotive

SEG Automotive is a global automotive supplier specializing in the development and production of starter motors, alternators, and components for electric drivers [3]. It provides innovative solutions for internal combustion engines as well as electrified vehicles. SEG Automotive aims to contribute to sustainable mobility by delivering products that enhance the efficiency and performance of automation systems.

The company has a deep connection to the automotive industry's history. It was pioneer for the starter motor and generator all the way back in 1913 and 1914 [3]. Over the years, it has played a crucial role in driving technological advancements within the industry. The main goal of the company is to develop competitive products that align with the future of transportation.

The company has around 7,000 employees in 14 countries in the world's most important automotive markets. The global team combines the highest level of engineering and production expertise, fostering close collaborations across cultural and national boundaries.

In essence, SEG Automotive stands as a dynamic and forward-thinking force withing the automotive industry, maximizing its rich history, technological expertise, and global reach to drive sustainable and competitive solution for the future of transportation.

1.1.2 Matrix 42 platform

In the dynamic landscape of modern enterprise management, organizations are continuously challenged to improve efficiency, security, and employees' experiences. The Matrix42 platform provides solutions that automate and simplify business processes to improve productivity, security, control, and user experience.

In addition to the diverse functionalities of Matrix42, in SEG Automotive, employees use Matrix42 to initiate support tickets. Whether they encounter issues, require assistant with malfunctioning elements, don't find a file or seek additional access or role permissions, they can seamlessly navigate through Matrix42 to submit a support ticket. This streamlined process ensures that employees can promptly address their concerns and receive the necessary support within the organizational framework.

Once the employee opens the support ticket, the helpdesk team responsible for checking Matrix42 tickets, can then evaluate the ticket, and give the best solution for each problem or doubt.

In summary, besides the many other functionalities that Matrix42 offers, it provides a communication platform between the employee in need and the responsible team.

1.2 Problem

Given the large number of employees at SEG Automotive company, the employees support process called Matrix42, where the employees can open tickets reporting their problems, is an important part for the smooth operation of SEG Automotive. The company already employs modern methodologies in this regard, with nearly all processes being conducted through its online support platform, the Matrix42.

Despite the several advantages, it brings both to the support team responsible for coordination Matrix42 support tickets, and to the employees that need help, some complications arise regarding time.

Particularly, it presents difficulties for employees to wait for their problems to be solve and presents challenges for the support team because it requires time and availability. This combined with the multiple tasks that the support team is responsible to do, can present some difficulties for the employees who need urgent help.

1.3 Objective

The proposed solution is to implement an intelligent agent capable of responding to the most frequently asked questions in Matrix42. In addition to streamlining the clarification of doubts by providing an immediate response to employees, the automation of this process alleviates the workload on the local support team and helps the employees to solve their problems.

The main goal of this project is to assist employees with their issues, providing a faster, effective, and self-sufficient interaction for them. It is important to ensure proper functioning and scalability.

The main objectives to implement this solution are:

- Analysis of requirements;
- Analysis of the most common issues and challenges employees face;
- Analysis and design the manual and current process;
- Evaluation and determination of the most suitable model and architecture for the solution that aligns with the company's needs;
- Development and design the intelligent solution;
- Train and test the intelligent solution;
- Deployment of the solution on SEG Automotive and evaluate its success.

1.4 Project Schedule

This section provides a schedule for this project, highlighting the main tasks of it.

In the project management domain, developing and executing a well-structured project schedule contributes to the success of the project delivery. In today's dynamic and fast-paced business environment, the ability to plan, execute, and monitor projects within specified timelines is primordial.

The diagram was chosen as it provides a visual representation of the project timeline, providing the clarity of the plan. The diagram, highlighting in table1, offers a comprehensive overview of the entire project, allowing for quick assessments of progress, deadlines, and task dependencies.

The project is divided into five main tasks: Planning, Design and Development, Production and Implementation, Tests and Adjustments and finally the Evaluation task.

In the first section, **Planning**, an overview of the project is done. The objectives of the project and the methodology in use are defined. Also, in this section, the main technologies are explored and explained in more detail, creating the section state of art.

Second, on the **Design and Development** phase, the company requirements are analysed, to develop the intelligent agent with all the company needs.

Third, on the section **Tests and adjustments** the intelligent agent will be tested, and in case of errors, the appropriated corrections will be implemented.

Fourth, the **Production and Implementation** phase, the agent will be implemented in the company. This section will work directly with the employees to understand if the agent is helpful and well used. This section also has the final tests.

Finally, the **Evaluation phase** contains the final analyses of the agent performance and contains some recommendations or changes for future works.

Table 1. Schedule Planner

1	Planning	Start	End	Days
1.1	Identification of Opportunity	15/10/2023	20/10/2023	6d
1.2	Objectives	15/10/2023	20/10/2023	6d
1.3	Research questions	15/10/2025	29/10/2023	15d
1.4	Research metodology	17/10/2023	31/10/2023	15d
1.5	Schedule Development	15/10/2023	31/10/2023	17d
1.6	Ethical considerations	19/11/2023	25/11/2023	7d
1.7	State of Art	01/11/2023	03/12/2023	33d
1.8	Technology assessment	01/11/2023	05/12/2023	35d
2	**Design and Development**	Start	End	Days
2.1	Design Requirements	02/12/2023	30/01/2024	60d
2.2	Software Requirements	02/12/2023	30/01/2024	60d
2.3	Company Requirements	02/12/2023	30/01/2024	60d
2.4	Data analysis	05/01/2024	28/02/2024	55d
2.5	Solution Design	06/01/2024	29/02/2024	55d
3	**Productions and implementation**	Start	End	Days
3.1	Solution Implemenation	05/04/2024	10/06/2024	66
3.2	Quality Control	10/05/2024	15/06/2024	39
4	**Tests and adjustments**	Start	End	Days
4.1	Implementation of tests	22/07/2024	10/08/2024	19
4.2	Documentation of test results	22/07/2024	20/08/2024	29
5	**Evaluation**	Start	End	Days
5.1	Project Performance Assessment	02/06/2024	10/07/2024	39d
5.2	Post-Implementation Review	02/06/2024	10/07/2024	39d
5.3	Recommendations Future Projects	18/05/2024	01/08/2024	76d

1.5 Document structure

The current document is divided into 9 chapters, each relevant to the overall understanding of the work developed. The current chapter provides a brief introduction to the project, exploring its context, the problem it aims to solve, the objectives, company background, and application.

The **chapter 2** is the state of art. This chapter gives an overview of the methodology used. The chapter also covers a systematic literature review process focused on implementing an intelligent agent (LLM) to enhance employee satisfaction and reduce the workload on support teams. The study follows a rigorous selection process for relevant research papers, narrowing down a large pool to a core set of studies that inform the project's outcomes and recommendations.

The **Chapter 3**, includes the value analysis and design of the project, including the needed value-added to the involved departments and the design of the project that includes the functional and non-functional requirements.

The **chapter 4** includes the flowchart of the project, and all the needed steps to develop the solution. Contains also the selected models for the solution and the results obtained.

The **chapter 5** contains the ethical considerations of the project.

Finally, **chapter 6** contains the overall conclusions of the project and some recommendations and changes to be implemented in the future, that can improve and add value to the presented solution.

2 State of art

This section introduces crucial concepts, gives a technological overview, and establishes an understanding of different technologies.

2.1 Research methodology

Figure 1 illustrates the visual representation of the process to guide the process development, outlined by Peffers et al. 2007 [4], which is interactive, flexible, and is divided in six phases.

1. Identification and Motivation of the problem;
2. Establish the objectives of a solution;
3. Design and Development;
4. Demonstration;
5. Evaluation;
6. Communication.

The selection of the Design Science Research Methodology for this dissertation stems from its customized approach to solving specific organizational challenges through the systematic creation and evaluation of IT artifacts. By using the Design Science Research Methodology, the dissertation aims to go beyond traditional frameworks, it emphasizes the proactive development of tangible artifacts, such as innovative technological solutions, to address real-world problems faced by organizations. This approach not only provides a deep understanding of the current problem, but also directly contributes to the further development and improvement of organizational processes.

The iterative nature of the Design Science Research Methodology ensures that the development and evaluation of IT artifacts are conducted in a continuous cycle.

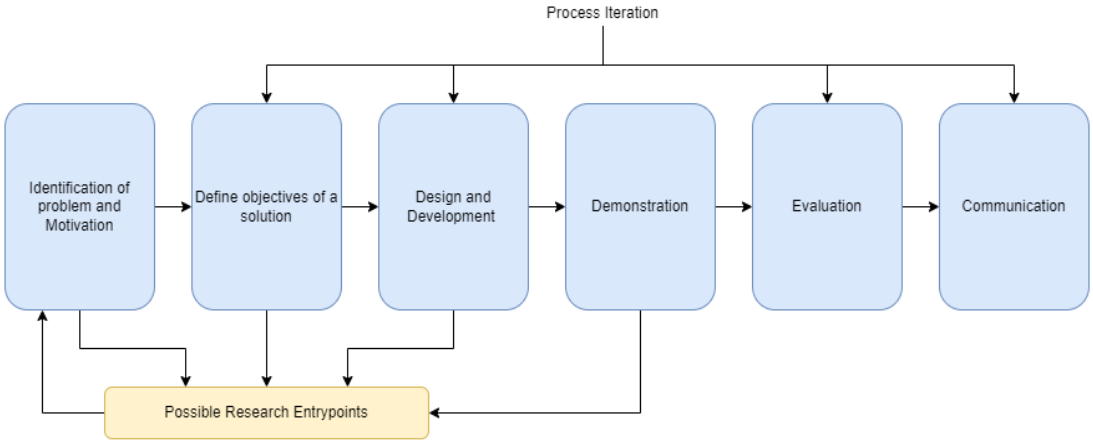

Figure 1. Design Science Research Methodology [1]

The first step**, identification of the problem and motivation**, consists into a clear definition and analysis of the helpdesk application, Matrix42 and the importance of finding a solution to help the support team.

The second step, **define objectives of a solution**, the goal is to evaluate what needs to be improved and the objectives related to improve the manual process and subsequently implementing automation. The thirst step is the **design and development**. In this phase, an architectural solution should be formulated, and the project plan must be finalized. In the development phase, the developer builds the intelligent agent algorithm based on the specific requirements, functional documentation, and the project plan established during the design phase. After this, the focus on the **demonstration** step should lies on handling process exceptions. On the fifth step, **Evaluation**, it is crucial to conduct a comparison to assess the impact of automation by contrasting the solution's performance with and without its implementation.

The six, and final step is **communication**, which plays an important role in this project. In this concluding phase, the emphasis is placed on effectively conveying information, insights, and findings derived from the entire process. Through this project, comprehensive and articulate communication will be employed to share the outcomes, implications, and pertinent recommendations that have arisen throughout the course of the activities undertaken.

2.1.1 Paper selection Process

In accordance with the previous described framework and relevance, this study aims to investigate and implement an agent for interaction with employees. The central question for this problem can be stated as:

1. **Main Question:** How can a new intelligent agent increase employees' satisfaction and reduce the workload of local support team?

This study has been undertaken as a systematic literature review based on the original guidelines as proposed by Kitchenham [5], an emeritus professor within the School of Computing and Mathematics at Keele University. The steps used in this methos are documented below:

1. **Data sources**
   The following Data sources were selected: Information and Software Technology (IST), Journal of Systems and Software, IEEE Transactions on Software Engineering, IEEE Software, Communications of the ACM (CACM), Journal of Artificial Intelligence Research (Jair), artificial Intelligence Review (Springer), Journal of Machine Learning Research (JMLR), ACM Computing Surveys (CSUR) and Proceedings of the International Conference on Learning Representations (ICLR).

2. **Research Questions**
   In the Kitchenham, method, after defining the data sources, the next step is to define the main research question of this project. These questions present the main topics/questions of the current project.

   The research questions addressed by this study are:

   - **RQ1**: What is an LLM?
   - **RQ2**: Why are LLMs so important?
   - **RQ3:** What are the technologies needed for implementing an LLM?
   - **RQ4**: What are the main types of LLMs?
   - **RQ5**: What are the benefits and challenges of implementing an LLM?
   - **RQ6**: How can the implementation of an LLM enhance user's interactions and satisfaction?
   - **RQ7**: What ethical and legal factors should be considered when incorporating automation solutions?

The **first question** focusses on the meaning of what an LLM is, the **second question** focus on the importance of having an LLM in organizations. The **third question** focusses on the needs behind the implementation of an LLM. **Fourth question** focusses on the main existing types of LLMs. The **fifth question** is about the benefits, and challenges faced when implementing an LLM. The **Sixth question** focus on employee's satisfaction, evaluating if the implementing of the intelligent automation is a good solution. Finally, the **seventh question** is about the ethical and legal factors that must be considered when implementing the intelligent agent.

3. Search terms

To transform the previous research questions into sets of words or queries, which can be used in databases to obtain the right results. The search terms were applied to abstract, title, and keywords of the paper. With the search words, combinations were attempted and evaluated over the number of results that they can provide before the final search query was reached.

Table 2. Domains and Keywords used to select search terms

Domain	Keywords
LLM	("Natural Language Model" OR "LLM" OR "AI Language Model" OR "Text generation Model" OR "Transformer Model")
Artificial Intelligent	("Artificial Intelligent" OR "Machine Intelligence" OR "Cognitive Computing" OR "Automated Intelligence" OR "Smart Technology")
Machine Learning	("Machine Learning" OR "Automated Learning" OR "Computational Intelligence" OR "Deep Learning")
Intelligent agent	("Intelligent Agent" OR "Autonomous Agent" OR "Cognitive agent" OR "Smart Agent" OR "Responsive Agent")

4. Data Extraction

Table 2 shows the String Query used in this project to search for papers on this theme. The query contains all the terms needed to build a trusted and reliable project.

Table 3. Query String

("Natural Language Model" OR "LLM" OR "AI Language Model" OR "Text Generation Model") OR ("Transformer Model") OR ("Artificial Intelligent" OR "Machine Intelligence" OR "Cognitive Computing" OR "Automated Intelligence" OR "Smart Technology") OR ("Machine Learning" OR "Automated Learning" OR "Computational Intelligence" OR "Deep Learning") OR ("Intelligent Agent" OR "Autonomous Agent" OR "Cognitive agent" OR "Smart Agent" OR "Responsive Agent")

To evaluate if a paper should be included in this project or not, a set of rules were devised. These rules will evaluate each paper using the criteria of inclusion and exclusion.

- The inclusion criteria must contain all the rules that the papers of this article must follow.
- The exclusion criteria contain all the exclusion criteria, it means, everything that will be rejected when the analysis of a paper comes.

Table 4. Inclusion Criteria

Inclusion Criteria	
IC1	The source belongs to the field of knowledge engineering/representation in computer science
IC2	The source describes the meaning of an LLM
IC3	The source describes existing types of LLMs
IC4	The source describes the importance of LLM
IC5	The source describes benefits and challenges of implementing an LLM
IC6	The source describes the technologies needed for implementing an LLM

Table 5. Exclusion Criteria

Exclusion Criteria	
EC1	The source is over 10 years old
EC2	The source is not written in English
EC3	The source focuses on a specific domain.
EC4	Contribution to the theme of this paper is unclear or not the focus of the source
EC5	The source doesn't specify the theme of this paper, only mentions the topic, without going deeper or explaining them in detail.

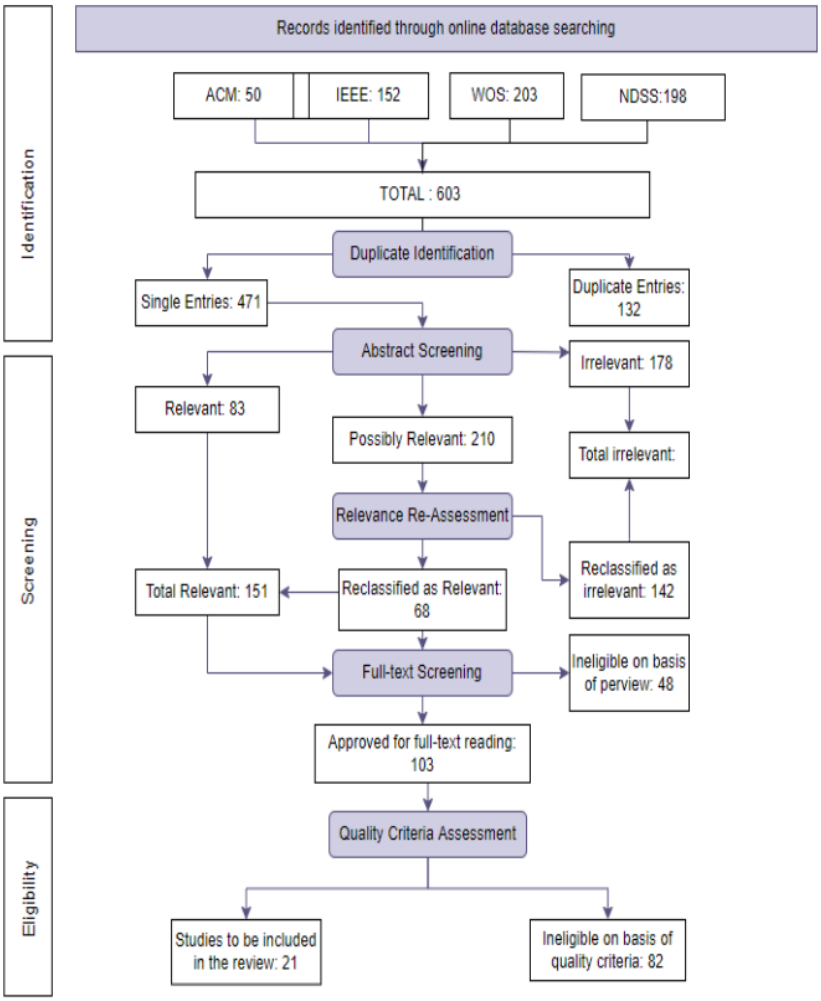

Figure 2. Flow Diagram showing the eligibility process

As seen in figure 2, there were 603 initial papers. At the end, with all the filters applied, there were a total of 21 studies to be included in the review.

From the 21 studies, the following were included and serve as the cornerstone, providing the foundation for the comprehensive analysis and insights that follow along this paper: [6] [9] [10] [11] [12].

At the end, after the LLM (Intelligent Agent) implementation, the following questions can be made, to evaluate the final solution.

- **RQ1:** What challenges does the current technical support system face at SEG Automotive Portugal, Lda?

- **RQ2**: How can Artificial Intelligence, Natural Language Processing, and Large Language Models be effectively integrated into the helpdesk operations to improve efficiency and user satisfaction?

- **RQ3:** What are the most effective methods for training an intelligent agent to handle common technical issues reported by employees?

- **RQ4:** What are the practical steps for implementing an intelligent agent using different LLMs?

2.2 Artificial intelligence, Machine learning and Deep learning

Artificial intelligent, Machine learning, and Deep learning are frequently employed interchangeably, but they do not entirely refer to the same concepts. Figure 3 highlights the fundamental differences between artificial intelligent, machine learning and deep learning.

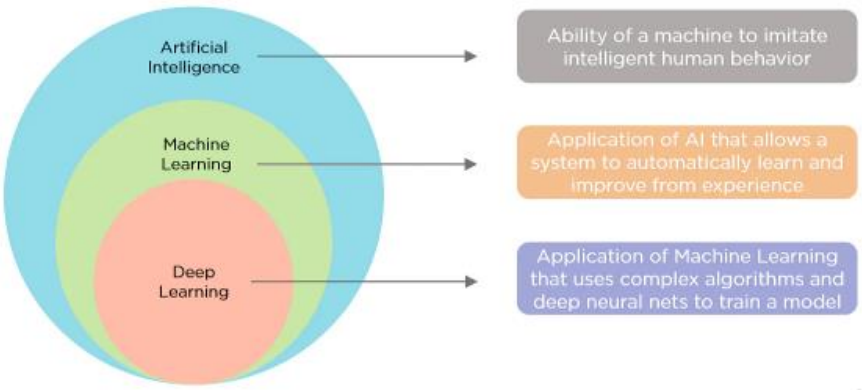

Figure 3. Artificial Intelligence Scheme [6]

Figure 3 illustrates the hierarchical relationship between three key concepts in the field of computational intelligent. **Artificial intelligence** represents the broadest category within the hierarchy. It encompasses the entirety of effort and technologies aimed at enabling machines to replicate intelligent human behavior. **Machine Learning** is a subset within AI. It is a subfield of AI that focuses on the development of algorithms and statistical models that allow computers to improve their performance on a tack through experience data. This adaptive learning process allows for continuous improvement of models based on incoming data. **Deep learning** is a subset of machine learning. This field involves the use of complex, multi-layered neural networks that are capable of automatically learning and extracting high-level features frow raw data [7].

In the next section, these three concepts will be explored in more detail.

2.2.1 Artificial Intelligence

Artificial Intelligence, commonly referred to as AI, is a branch of computer science focused on developing intelligent computer systems capable of perceive, analyze, and respond accordingly to various inputs [8].

One of the primary objectives of AI is to create autonomous machines capable of thinking and behaving like a human. These machines can replicate human behavior, executing tasks through learning and problem-solving mechanisms. Most AI systems emulate natural intelligence to address complex problems [8].

2.2.2 Machine learning

Machine learning is the foundations for most AI solutions. From the 1950s onward, researchers, often referred to as data scientists, have explored various AI methodologies. Many modern applications of AI have their origins in machine learning, a field that merge computer science and mathematics [9].

In the words of Arthur Samuel, machine learning is characterized as the field of study that empowers computers to learn without requiring explicit programming. Machine learning (ML) is employed to instruct machines on how to manage data more efficiently. In situations where the information extracted from the data cannot be interpreted, machine learning comes into play. The increasing availability of datasets has led to a growing demand for machine learning. Many industries apply ML to extract relevant data. In summary, one of the main purposes of ML is to learn from the data [9].

Machine learning relies on different algorithms to solve data problems. Data scientists like to point that the kind of algorithm employed depends on the kind of problem to solve.

Figure 5 shows how a machine learns from data.

Figure 4. Machine Learning flow adapted [10]

As illustrated in figure 5, the Past Data, represents historical data collected from various sources. Then, the middle section illustrates the machine learning model training phase, where the model analyses and learns patterns from the past data. Then, on a third phase the trained model is used to make predictions or generate outputs on new input data.

In today's world, people generate vast volumes of data as they engage in their routines. From sending text messages, emails, and social media posts to capturing photographs and videos on their phones, individuals produce substantial amounts of information. Additionally, millions of sensors in homes, cars, cities, public transport infrastructure, and factories contribute even more data.

Using all the historical data, data scientists can train ML models to make predictions and draw inferences by identifying relationships within the data and learning form the past data. The models aim to capture the connections between data.

Advancements in AI methodologies have progressed to complete tasks of significantly higher complexity, forming the cornerstone of AI capabilities.

2.2.2.1 Algorithms

The functionality of an ML algorithm relies on two key concepts: the action being learned, known as the label, and the corresponding features that delineate the justifications for labeling. These features are composed of one or more attributes. Methodologies are categorized into three types based on their learning approach: supervised learning, unsupervised learning, and reinforcement learning [11]. For this thesis, only the supervised learning will be explored in more detail, since it is the algorithm that will be applied in this project.

I. Unsupervised learning

In **unsupervised learning** there are no correct answers or teachers guiding the process. In this paradigm, algorithms learn few features from the data. When new data is introduced, it uses the previously acquired features to identify the data's class. This method is primarily employed for tasks such as clustering and feature reduction [11].

II. Reinforcement learning

Reinforcement learning consists of operational agents aiming to identify the best procedure to obtain a reward or improve performance over time. This approach relies on iterations categorized as either positive or negative, persisting until level is reached that aligns with the ultimate objective. The classification is cumulative, providing the agent with assessments of what it should or should not do after each training phase. Its learning process is delineated by three fundamental concepts: the learning policy, specifying the agent's behavior at a given moment, the reward signal, defining the learning goal, and the value function, a model that evaluates whether the agent's behavior is favorable for future iterations [12].

III. Supervised Learning

On the other hand, **supervised learning** involves the ML process of acquiring a function that maps input to output by leveraging example input-output pairs. This method deduces a function from labeled training data, comprising a set of training examples. These algorithms require external assistance, and the input dataset is typically divided into training and test datasets. The training dataset includes an output variable that requires prediction or classification [13]. All algorithms learn patterns from the training dataset and subsequently apply these patterns to the test dataset for prediction or classification.

There are two types of algorithms in Supervised Learning [13]:

- Classification – Assumes discrete values from a non-ordered dataset. (X's and O's)
- Regression – Assumes infinite values from an ordered dataset.

Figure 5 highlights the difference between these two types of algorithms.

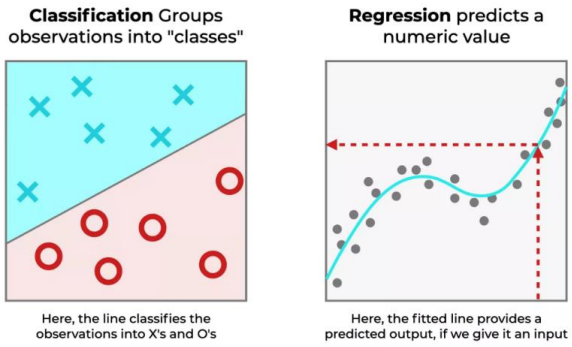

Figure 5. Classification vs Regression Algorithms [14]

In conclusion, the choice between unsupervised, reinforcement, and supervised learning models depends on the specific requirements and nature of the task at hand. Unsupervised learning is ideal for exploratory data analysis and situations where labeled data is not available. Reinforcement learning is best suited for tasks involving sequential decision-making and environments that provide feedback. Supervised learning excels in scenarios where labeled data is available and high accuracy in prediction or classification is needed. Each model has its unique advantages and limitations, and selecting the appropriate one is crucial for the success of a machine learning project.

2.2.3 Deep Learning

Deep learning is a subfield of machine learning that focuses on algorithms that draw inspiration from the structure and functionality of the human brain. These algorithms have the capacity to handle enormous quantities of both structured and unstructured data. Deep learning's core concept is the utilization of artificial neural networks, empowering machines to make informed decisions [15].

The key distinction between deep learning and machine learning lies in the way data is presented to the system. ML algorithms typically demand structure data, and the system is mostly put either in supervised or unsupervised learning method with multilayers of algorithms undergoing such learning methods. With the increase of number of layers, it is called as deep learning or Deep Neural Network. In summary, ML algorithms typically demand structured data, whereas deep learning networks operate through multiple layers of artificial neural networks.

Artificial Neural Networks (ANN) are computational systems designed to copy the biological operations of the human brain. They can be conceptualized as graphs with several levels, internally consisting of nodes connected by mathematical weights. Neural networks comprise three layers: the input layer, the output layer, and the hidden layer, which may have a variable number of internal levels [16].

Figure 6 shows the workflow of a deep neural network.

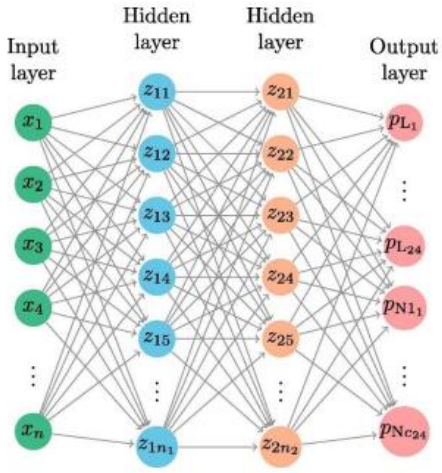

Figure 6. Deep Neural Network [16]

The **deep learning flow** consists in five steps [17]:

1. Calculate the weighted sums.
2. Pass the computed sum of weights as input to the activation function.
3. The activation function takes the "weighted sum of input" as its input, incorporates a bias, and determinate whether the neuron should activate.
4. The output layer produces the predicted output.
5. Compare the model's output with the actual output. Post-training, the neural network employs the backpropagation method to enhance its performance. The cost function helps minimizing the error rate.

The fundamental concepts of a neural network are the cell and its corresponding activation state, the connections between units, the activation function that defines a cell's output, the cost function, and, ultimately, the method of information gathering [18].

The role of the cost function is to systematically compute the difference between the obtained result and the expected outcome, indicating, though a numerical value, whether the weights of each connection should be increased if positive or decreased if negative. Taking these values into account, the adjustment of the neural network is achieved through the backward propagation algorithm.

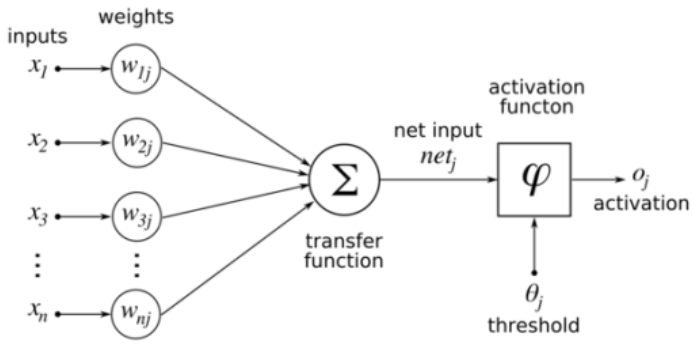

Figure 7. Internal functioning of a neuron [18]

Based on execution orientation, there are two types of neural networks.

I. **Feed-forward Networks** operates in a single direction from input to output.
II. **Recurrent Neural Networks (RNN)** are characterized by neurons having cyclic connections to previous ones [18].

Despite the advantages of RNN, the main issue lies in the fact that they suffer from the vanishing and exploding gradient problems. Several variants of RNNs have been proposed to alleviate this issue using various mechanisms. The most common ones are [18]:

- **Long-Short-Term memory units (LSTMs)** are used extensively in NLP. Introduced in 1997 by Hochreiter and Achmidhuber, recurrent neural networks have gained growing popularity in recent years, thanks to advancements in hardware accelerated deep learning. Additionally, they have demonstrated promising outcomes in fields such as machine translation and image captioning.

  The memory cells are the points of awareness that decide which data to preserve, using gates, particularly the input gate, responsible for determining whether new information should be stored, the output gate, tasked with sharing or not sharing the memory with the neural network, and finally, the forget gate, assigned to erase the existing memory.

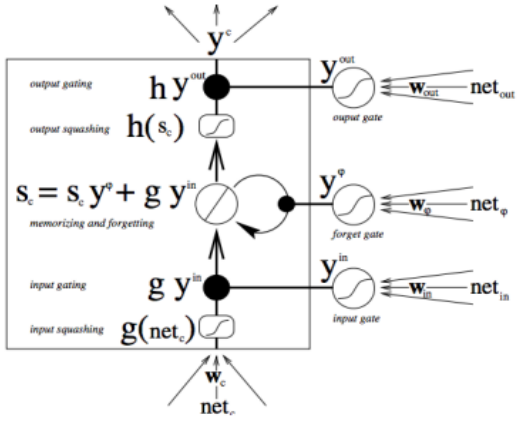

Figure 8. Internal behaviour of LSTM units [18]

As observed in figure 8 the neural network's information is directed to the three constituent elements of the LSTM unit, and each will have its influence on the output to be sent.

- **Gated Recurrent units (GRUs)**, represents a variation of LSTMs, combining the forget and input gates into a single gate, along with merging the cell and hidden states. This consolidation leads to a reduction in the number of parameters to be tuned, resulting in shorter training times for GRUs compared to LSTMs. Nevertheless, their performance has been demonstrated to be comparable to that classic LSTM networks.

2.3 Natural language processing

Data preparation phase comprises natural language processing (NLP) methods, that aims to transform the received content into a structured and suitable textual format. The objective is to ensure that the data is in a predictable and conductive form for subsequent analysis and modeling within the context of deep learning.

2.3.1 Scraping

The scrapping technique consists of extracting information automatically from a digital environment full of data in a systematic and efficient way. After data extraction, the information is converted into content in an accessible format and ready for later analysis [19].

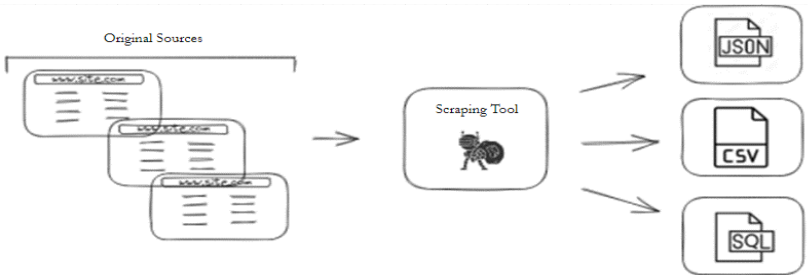

Figure 9. Scraping process

Figure 9 highlights the transformation of original sources, containing several information's into a specific and accessible formats like Json, CSV and SQL.

2.3.2 Data pre-processing

In the scenario of data analysis and NLP, the text pre-processing technique emerges as a primary step for cleaning data and removing noise before the data is processed by ML algorithms and models. By carrying out this procedure on the text, it is possible to guarantee an improvement in the quality and accuracy of any analysis throughout the project. Furthermore, by organizing textual data into a uniform, easy-to-understand format, preprocessing allows machine learning models to learn patterns more efficiently [20].

Tokenization is the process of diving text into smaller structures known as tokens. Its primary objective is to transform sentences into individual elements, utilizing spaces, punctuation marks, or special characters for separation. Example:

- Initial Phrase: Where can I open a Ticket?
- Can be divided in 7 tokens: Where + can + I + open + a + ticket + ?

In some cases, tokenization may compromise the meaning of a word, such as abbreviations, names containing punctuation, the use of conjunctions with pronouns, and even language-specific nuances. Nevertheless, tokenization algorithms are generally well-adapted to the nuances of most languages.

The process of **eliminating stopwords** entails the removal of connecting words that don't substantially contribute to the meaning of a sentence. Typically, these include grammatical classes like pronouns, articles, conjunctions, and prepositions. As observed on table 6, despite the reduction, the final sentence is more understandable than the original one. Besides enhancing processing efficiency, the removal of these words contributes to the clarity of the intended meaning [20].

Table 6. Elimination stopword

Phrase	Stopwords			Final
Was the ticket opened in Matrix42?	was	the	in	Ticket opened Matrix42?

Finally, for the goal of standardizing their meaning, there are algorithms designed for morphological manipulation of words. Morphology is related to the meaning, structure, and relationship between words.

There are two types of morphological transformations, the conversion of a word into a stem and the conversion of a word to a lemma.

The **Stemming** is the procedure of reducing inflected or derived words to their base or root form, usually a written word form. It is defined as the removal of suffixes and prefixes from words to obtain their root [20].

The **lemmatization** is an alternative to the stemming method, using dictionaries and morphological analysis of words to convert them into canonical form [20].

2.3.3 Methods of meaning extraction

- **Chunking**

  The chunking technique plays an essential role in decomposing and understanding the semantic units contained in texts more effectively. It consists of segmenting the text into smaller and more cohesive parts, where each of these pieces represents a semantic unit that can consist of one or more words, forming blocks of more coherent meanings [21].

  One of the main reasons for carrying out chunking is to facilitate semantic analysis and information extraction, since in longer texts, information can be dispersed in several parts of the text. This way, the technique helps to identify the words that are related and in identifying parts of the text that are relevant to analysis. Therefore, by dividing a text into meaningful pieces, it is possible to capture the relationships between words and concepts, giving the possibility to carry out a more precise and in-depth analysis [21].

- **Embedding**

  The embedding technique is a foundational method in NLP that transforms individual words into numerical representations, known as vectors, within a multi-dimensional space. Each word is assigned a unique vector comprised of real numbers, typically with dozens, hundreds, or even thousands of dimensions. These vectors are not arbitrary, rather, they are structured in such a way that the position and orientation of each vector in the space encapsulate the semantic properties of the corresponding word [22].

  The essence of this technique lies in its ability to preserve and convey semantic meaning through the geometric relationships between vectors. Words that share similar meanings or are used in similar contexts tend to be mapped to vectors that are positioned closely together within this vector space. For example, the words "king" and "queen" would be represented by vectors that are close to one another because they share similar semantic attributes. Conversely, words with vastly different meanings, such as "king" and "computer," would be mapped to vectors that are far apart [22].

  This spatial arrangement allows for the measurement of distances between vectors, which can be interpreted as a measure of semantic similarity or difference. By analysing these distances, algorithms can detect patterns, relationships, and nuances within the language. For instance, the difference in vectors between "man" and "woman" might be like that between "king" and "queen," reflecting a consistent semantic shift.

The power of embedding lies in its capacity to enable machine learning models to process and understand human language in a more sophisticated and contextually aware manner. These models can leverage the structured relationships within the vector space to perform tasks such as sentiment analysis, machine translation, and question-answering with improved accuracy. By capturing the subtle relationships between words, embeddings facilitate a deeper comprehension of language, allowing machines to engage more effectively with the complexities of human communication [22].

2.4 Large Language Models

LLM are learning models that represents a form of AI capable of comprehending and generating natural language text. Through exposure to extensive datasets sourced from materials such as books, articles, web pages, an LLM discover patterns and rules of language [23].

An LLM is built through a neural network architecture. This architecture takes an input, has several hidden layers which break down in various aspects of language, and generates the output layer. In short, the more parameters a model has, the more data it can process, learn from, and generate [23].

In **traditional NLP**, distinct models are typically required for each specific capability or task. These models are trained using labeled data, where each input is associated with a corresponding output or category. The training process often involves describing in natural language what the model is intended to accomplish [24].

On the other hand, with **LLMs**, a single model is utilized to address multiple natural language use cases. Instead of relying solely on labeled data, LLMs are trained on vast amounts of unlabeled data, often spanning many terabytes, to establish a foundational understanding of language. Also, LLMs are highly optimized for various specific use cases, enabling them to adapt to a wide range of tasks without the need for separate models [24].

There are a few core concepts that are important to understand to effectively use LLMs: prompt Engineering and Token.

2.4.1 Prompt Engineering

Prompt engineering is a technique that refers to formulating prompts, that is, specific instructions provided to language models with the aim of obtaining responses in the desired way. This is a crucial step in the use of language models, such as in text generation tasks, where AI models generate automatic responses based on the input received. Therefore, the prompts are important so that the quality and relevance of the answers obtained through the language models are cohesive, meeting the user's expectations [24].

2.4.2 Token

A token represents a single character, a portion of a work, or even an entire word. Common words may be represented by a single token, whereas less frequent words might require multiple tokens [24].

A token is a fundamental unit of text that an LLM can understand and process. OpenAI’s natural language models don’t use words or characters as units of text [25] . Instead, they operate on tokens, which are units that fall somewhere in between.

Table 7 shows an example of the number of tokens in some words.

Table 7. Example of tokens

Word	Number of Tokens
Apple	1
Blueberries	2 (Blue + berries)
Skarsgard	Multiple Tokens

The token representation on table 7 is what enables AI models to produce words not found in dictionary, without needing to generate text letter by letter.

2.4.3 Generative AI

As AI technology continues to evolve, generative AI has emerged as a key focus of research. According to Lim et al. (2023) [26], generative AI is an innovative technology capable of automatically creating new content from input data. Its theoretical framework is built on machine learning, natural language processing (NLP), image processing, and computer vision.

Generative AI models can be categorized into three types: Natural Language Models, Generative pre-trained transformer and image generation models.

First there is the **Natural Language models**, which process natural language inputs and generate responses. They can perform a variety of natural language tasks, such as: summarizing text, classifying text, generating names or phrases, answering questions, translation and suggest content [27].

Second, **Generative pre-trained transformer** (GPT), developed by OpenAI, are a set of neural networks that use the transformer architecture used for NLP tasks and offer solutions and the ability to generate content like humans by generating a readable and coherent response.

The Transformer architecture is a neural network architecture designed for natural language processing. It consists of an encoder and a decoder that work together to generate natural language text as shown in figure 11.

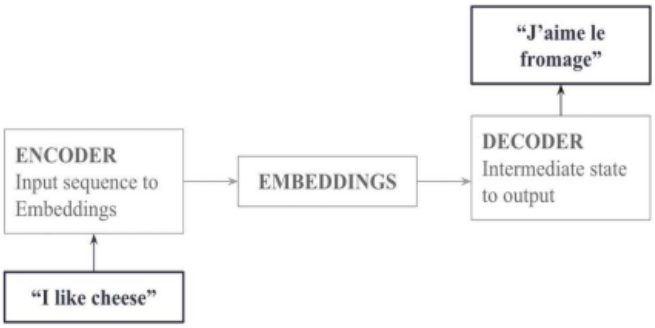

Figure 10. GPT-3 Transformer Architecture Diagram [7]:

These models are pre-trained on large amounts of data, using scraping techniques and other content sources. These models can take natural language or code snippets and translate them into code. These models are proficient in programming languages, including C#, JavaSript, PHP, Perl and Python. Using LLMs for coding helps address several challenges [7]:

- Building applications: LLMs can generate code for projects like web APIs based on given prompts.
- Maintaining applications: Assist in updating or maintaining an existing codebase.
- Improve application: Enhance code to meet specific metrics, such as increased security or improved logging.

And lastly**, Image generation models**, which use prompts, a base image, or both to create new images [27].

2.4.4 Training methods

Pretraining differs from the traditional backpropagation approach in neural networks, which typically starts with randomly initialized parameters. Instead, pretraining involves training a model on specific tasks to acquire pretrained parameters. These parameters are then used to initialize the model for subsequent fine-tuning. Pretraining is a technique within the broader field of transfer learning [28].

The first-generation of pretrained model emerged in 2013 with the introduction of word2vec, mention above in this document, which provides word representation for training neural networks. In 2018, ELMo marked the beginning of the second-generation of pretrained models, adopting the "pretraining + fine-tuning" paradigm [28].

Later, the more advanced Transformer architecture was utilized in subsequent pretraining language models such as GPT and BERT, consistently achieving state-of-the-art performance in natural language processing task.

During the **pretraining phase**, the model learns from a vast and diverse dataset, which can be broadly categorized into general and specialized data. General data, such as web pages, books, and conversational texts, are commonly used because of their large scale, diversity, and accessibility. This phase allows the model to learn general language patterns and representations, thereby improving its language modelling and generalization capabilities [28].

In the **fine-tuning** stage, the model is further trained on smaller, more specific datasets related to the target domain or task. This process integrates the pretrained model's generalization abilities with the requirements of the target task, therefore improving performance. Fine-tuning typically involves freezing certain parameters, updating top-level parameters, and adjusting others. As soon as fine-tuning is complete, the model is ready for deployment in practical applications for specific tasks [28].

2.4.5 Recent models

In this section LLM models are presented.

A. ELMo (Allen Institute)

Elmo is a “deeply contextualized word representation thar captures the complex features of word usage and variations of these usages are modeled across linguistic contexts” [29].

Deep neural networks aim to create high dimensional representations of words that consider their context and capture their syntax and semantics. The network is trained on large text datasets and sequentially predicts the next word based on previous words, allowing it to understand complex relationships between words and meanings [30].

This LLM uses long, short memory in its two-layer bidirectional architecture, along with the character-level convolutional neural network (CNN) [30].

The CNN consist of layers of nodes, including an input layer, one or more hidden layers, and an output layer. Each node is connected to others with associated weights and thresholds. When the output of a node exceeds its threshold value, the node activates and passes data to the next layer; otherwise, no data is transmitted to the following layer [30].

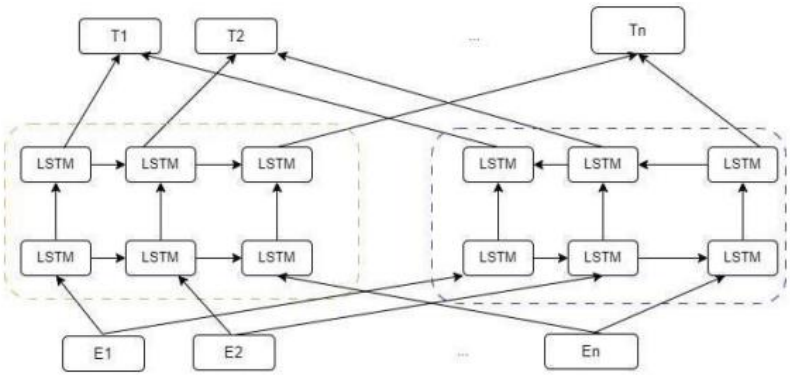

Figure 11. ELMo Architecture [31]

As shown in the figure 11:

- The sequence of sentences is input to ELMo, which then feeds into the CNN.
- The CNN generates raw word vectors, extracting meaning from the text data and representing each word as unique vector. These vectors serve as input for the bidirectional language model (BiLM) – LSTM [29]
- The BiLM processes the sequence of words in both forward and backward directions. The backward LSTM captures information about each word and its subsequent context, while the forward LSTM captures information about each word and its preceding context [29].
- LSTM is a type of neural network specifically designed for sequential data. It addresses the vanishing gradient problem found in traditional RNNs by incorporating “memory cells” that retain information over long periods and “gates” that regulate information flow. The input gate allows new information in, the forget gate discards unnecessary information, and the output gate determines the output information.
- LSTM can remember past inputs and generate outputs based on them, effectively capturing dependencies between words or other features in a sequence.

B. Falcon

Falcon is an advanced open-source language model designed to handle a wide array of text processing tasks. These tasks include, but are not limited to, content creation, where the model can generate high-quality written material, solve complex problems, make it a valuable tool for tackling intricate challenges, serving as a virtual assistant [32].

The most powerful variant of this model is Falcon-40B. As indicated by its name, this model is built with 40 billion parameters, making it highly sophisticated and capable of understanding and generating nuanced text. Falcon-40B was trained on an extensive dataset consisting of 1 trillion tokens, sourced from a specialized database called RefinedWeb, which was meticulously curated by the same development team. This extensive training allows the model to have a deep understanding of language and context, resulting in highly accurate and contextually relevant outputs.

One of Falcon's standout features is its open-source nature, which means that it is freely available for use, modification, and distribution. This openness not only facilitates commercial use but also democratizes access to advanced AI capabilities, enabling developers, researchers, and businesses to integrate Falcon into their projects without the constraints of licensing fees. Users can build upon the Falcon model to create custom applications, ranging from personalized virtual assistants to innovative AI-driven solutions, ensuring that responses are efficient, accurate, and tailored to specific needs. This open-source accessibility encourages a broad range of innovations, fostering a community-driven approach to further enhancing and applying the Falcon model in diverse domains [32].

C. GPT-3 (OpenAI)

GPT-3 (Generative Pre-trained Transformer 3) is an advanced language model. It is based on the transformer architecture and was trained using large amounts of data to generate human-like responses in natural language. Released on June 11, 2020, it has received significant attention from researchers, industry experts, and the media due to its advanced capabilities and potential for future application [33].

With 175 billion parameters, the model is the largest language model ever built. It is trained on a variety of tasks, including language generation, language translation, question answering, sentimental analysis, summarization, chatbots, and text completion. GPT can generate coherent and fluent text that is very similar to human writing [33].

This model uses a technique called attention, which allows the model to focus on specific parts of the input sequence when generating output. This technique is crucial for generating coherent and smooth text because it allows the model to understand the context of the text it generates. The model GPT-3 also is trained using large amounts of text data from the internet [33].

The model is assumed to be trained on 8 different model sizes ranging from 125 million to 175 billion parameters to study the dependence of machine learning performance on model size. Their goal was to determine if the scaling of validation loss follows a smooth power law in relation to size, which they examined for both validation loss and downstream language tasks [33]

D. Llama (Meta AI)

Llama stands for large language Model Meta AI – Metas LLM for Artificial Intelligence. It is open source and completely free, so researchers, government organizations, and even common users can access it for free. According to Meta's announcement, the model has up to 65 billion parameters and can achieve next word prediction in text sequences trained on text from 20 languages using Latin and Cyrillic alphabets [34]. It is claimed that the largest model was trained on 1.4 trillion tokens and the smallest model was trained on 1 trillion tokens. The model shows promise in more complex tasks such as generating text, chatting, summarizing written documents, and solving parametric mathematical theorems.

Llama is implemented on the transformer architecture, where the encoder encodes the input into a vector representing its semantics, and the decoder converts this vector into the target language. [34].

For the current project the four models: Llama, GPT-3, Falcon, and Elmo will be used. Each model will be systematically tested to assess its performance and capabilities in various contexts. This comprehensive evaluation aims to provide a thorough understanding of how each model performs and to compare their effectiveness in handling the tasks at hand.

2.4.6 Vector Embeddings

Although AI's language processing abilities have been improving over time, the need for numerical representation of textual data persists. Vector embeddings provide a numerical representation of various types of data [35]. For this thesis two text embeddings were used, since they transform text data into numerical representations:

I. OpenAI's Text Embedding-ada 002
II. Google's VertexAI's textembedding-gecko@001

Each segment of text is converted into a vector with a dimensionality of 1536 for OpenAI's text-embedding-ada-002 model and 768 for Google's textembedding-gecko@001 model. The two models differ in dimensionality, as mentioned, with lower dimensionality indicating that less information is stored [35].

Having multiple text embedding models enables to evaluate the effect of dimensionality on performance. Since both OpenAI and Google offer state-of-the-art embedding models, the results obtained from these models can be regarded as indicative of the overall performance of text embedding models.

After the text embedding model has transformed the text into a vector, the vector database conducts a similarity search on this vector.

In addition to the vector database's capability to classify data, it is crucial that the embedding model encodes the data in a way that accurately reflects its true meaning. If the embeddings do not represent the underlying content of the text, effective classification by the vector database becomes challenging. This is because poor encoding by the embedding model means that the vector representations fail to capture the text's meaning, making it harder to achieve accurate classification [35].

2.5 RAG

The Retrieval Augmented Generation (RAG) is a technique used in the NLP area that uses data recovery (retrieval) and text generation (generation) strategies [36].

Its purpose is to improve the quality and relevance of responses generated by language models.

Additionally, this technique allows a model to access information from a knowledge base, and, within this context, it is possible to generate more accurate and informative answers that are aligned with the original references to users' questions, bringing the most significant results [31].

As mentioned, the first part of RAG is about retrieving relevant information from a database. This stage is done through information search and retrieval techniques that aim to identify some similarities present in the text that may contain expressive information.

The second part of the RAG is about the response generation stage based on the information retrieved in the previous stage. For this, there are several LLMs, which are generally used in this phase to create coherent answers based on the context obtained [36].

2.5.1 Possible applications

As the field of NLP is quite broad, being able to act in different areas, the RAG technique has also become very effective in several applications, bringing several advantages in the following areas [36]:

1. **Improving the quality of answers**: Using context recovered from original references, RAG can deliver relevant and contextually enhanced answers, so that the results are able to fit within the scope of the question.

2. **Answering complex questions**: Since the user is not always able to make their doubts clear, the use of RAG can deal well with this type of situation, as it allows the model to access relevant information from a database, enabling the LLM to respond more precisely and coherently.

3. **Diversity of uses**: In addition to answering questions, RAG has been used in tasks such as translation, summarizing texts, creating chatbots, etc. Your ability to improve the quality of responses is valuable in a wide variety of scenarios.

Therefore, RAG emerged as a promising solution to overcome the limitations of LLMs. By using this technique, developers can obtain personalized solutions while maintaining data coherence, while making the most of LLM capabilities when generating results.

2.6 Open-Source tools

In this section, the resources used, and the methods applied will be specified. For the development of this study, it was necessary to use several resources such as the programming language and its libraries, available datasets for use, and natural language processing models.

This section presents and explores the tools used.

2.6.1 Hugging Face

Hugging Face is a comprehensive platform designed for the creation, training, and deployment of open-source Machine Learning Models. It facilitates collaboration among professionals and enthusiasts in the field, supporting the development of complete end-to-end applications.

One of the key strengths of this platform lies in its development and upkeep of libraries such as transformers. This ensures easy access to a diverge range of large-scale language models. These libraries streamline the implementation of tasks such as text generation, sentiment analysis, machine translation, and more. By making AI more accessible to developers of all experience levels, they significantly reduce both time and resource expenditure [37].

2.6.2 LangChain

LangChain is an open-source framework created by Harrison Chase, to build LLMs by chaining interoperable components, enabling the use of emerging technologies stemming from NLP [38].

This tool streamlines the development of a wide range of applications, including chatbot functionalities, automated question and answer generation, and summarization of specific materials. It effectively integrates components from various models, enabling the creation of sophisticated applications centered around an LLM.

2.6.3 Evaluation metric

Evaluation metrics play an important role in determining the effectiveness and overall success of a model. They provide a quantitative measure of the model's performance, allowing developers to assess how well the model meet its objectives. For this purpose, a metric was used during the project: BLEU Score.

The BLEU Score, which stands for Bilingual Evaluation Understudy, is a metric used to assess the quality of responses generated by language models. It was initially developed to evaluate translations but is also widely applied to broader natural language processing tasks [39].

This metric ranges from 0 to 1, with higher values indicating a greater match between the generated response and the original reference. The BLEU Score calculates the overlap of N-grams between the generated result and the original text. An N-gram is a sequence of N consecutives tokens, with commonly used values being 1, 2, 3 and 4, representing unigrams, bigrams, trigrams, and quadrigrams, where a token can be a word or a syllable. For each N-gram, BLUE assesses how many of these tokens generated by the result are present in the original texts. It then counts the correct N-grams to calculate precision. Finally, precision is computed as the ration of the number of N-grams in the generated text to the total number of N-grams in the original reference [39].

It is important to highlight that, when using quadrigrams to obtain the BLUE Score result, it is possible to obtain a more accurate result, as it will be able to capture, he context of the result with greater complexity by comparing it with larger sentence.

The BLUE Score formula is given by the following format [39]:

$$BP \cdot exp(\sum_{n=1}^{N} w_n log \cdot p_n)$$

Figure 12. BLUE Score Formula [39]

Where the value of P_n represents the value of N-grams and W_n is the value of the chosen N-gram.

3 Design

This section presents the design of the project, including the As-Is diagram and the To-Be diagram and the functional and non-functional requirements.

3.1 Design

In SEG Automotive Portugal's company all the support process are performed manually, with no usage of automation's during its duration. In each month an average of 300 requests for the support team is received. Figure 14 shows the actual process for support using Business Process Model and Notation (BPMN).

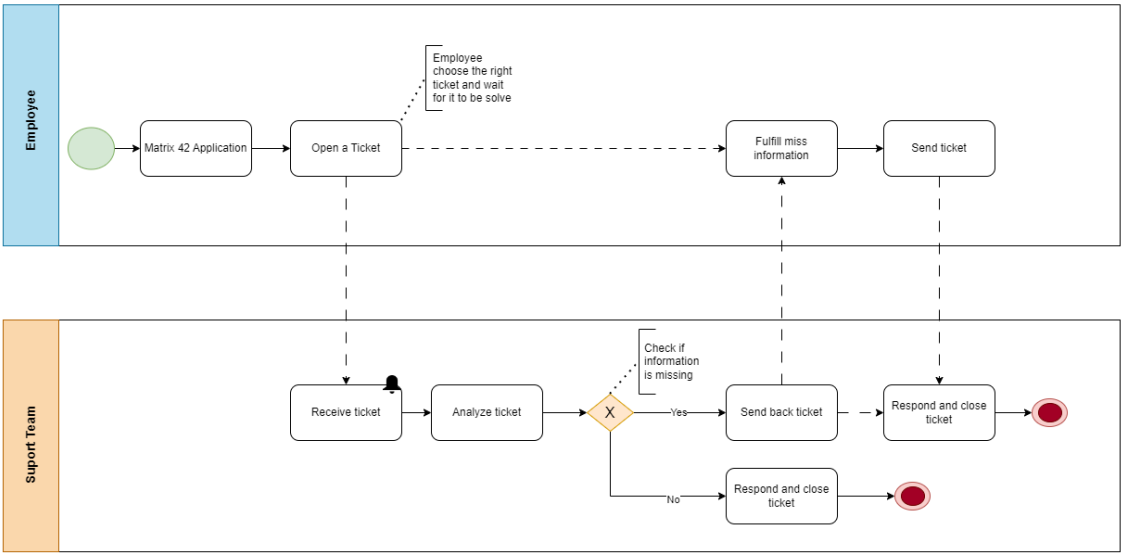

Figure 13. Ticket Request As-Is

In the figure 14 we can analyze that, first, if the employee needs support, he will have to open the Matrix42 application and choose the type of ticket corresponding to the problem he faces. After opening the ticket, the employee will be waiting for an uncertain period, depending on the availability of the support team.

In the second poll, Support team, we can analyze the following: when the support team responsible for answering and analyzing tickets in Matrix42 receives the notification of a new ticket, the team will analyze that ticket, depending on their availability. After the analysis, if any information is missing, the ticket will be sent to the responsible employee, who will fulfill the missing information, and send again the ticket, so then the support team can respond and close the ticket. On the other hand, if the ticket is correct, not missing additional information, then the support team will resolve the problem and close the ticket.

3.2 Requirements

After analyzing and understanding the process as it is made, a list of requirements has been made.

3.2.1 Functional Requirements

The functional requirements for the employee:

- **Matrix42 Application access**: employees must have access to the Matrix42 to initiate the ticketing process
- **Open a ticket**: Employees should be able to open a new ticket within Matrix42
- **Fulfill missing information:** Employees should be able to edit and update the ticket to fulfill any missing information as requested by the support team.
- **Send ticket**: Employees must be able to submit the completed ticket

The functional requirements for the support team:

- **Receive ticket**: The system should notify the support team of a new ticket submission
- **Analyze ticket**: The support team should be able to analyze and assess the ticket to determine the require actions
- **Check for missing information**: If information is missing, the system should facilitate sending the ticket back to the employee with a request for additional details.
- **Send back ticket:** The support team should be able to send the ticket back to the employee to request additional information
- **Respond and close ticket**: Once all necessary information is provided, the support team should be able to respond to the ticket, provide a solution and close the ticket.

3.2.2 Non-Functional requirements

In addition to the functional requirements, non-functional requirements were defined.

1. Performance:
    - Response time: The system should response to user actions within 4 seconds
    - Scalability: The system should handle up to 100 concurrent users without performance degradation.

2. Reliability:
    - Availability: The system should be available 99.9% of the time, ensuring minimal downtime.
    - Error handling: The system should gracefully handle errors and provide meaningful error message to users
    - Data integrity: The system must ensure data consistency and integrity

3. Usability:
    - User interface: The interface should be intuitive and user-friendly.
    - User training: Adequate training materials and help documentation should be provided to ensure users can effectively use the system

4. Security:
    - All data related to employ should be protected under the GDR.
    - The algorithm should use the data required for the process

3.3 To be

This To Be process represents the envisioned, optimized workflow that addresses current challenges, leverage new opportunities, representing the process as it is going to be executed after the automation is totally implemented.

The objective outlined in the to be process streamline the support process by allowing employees to interact directly with a Large Language Model for instant responses, reducing the need for manual ticket management by the support team.

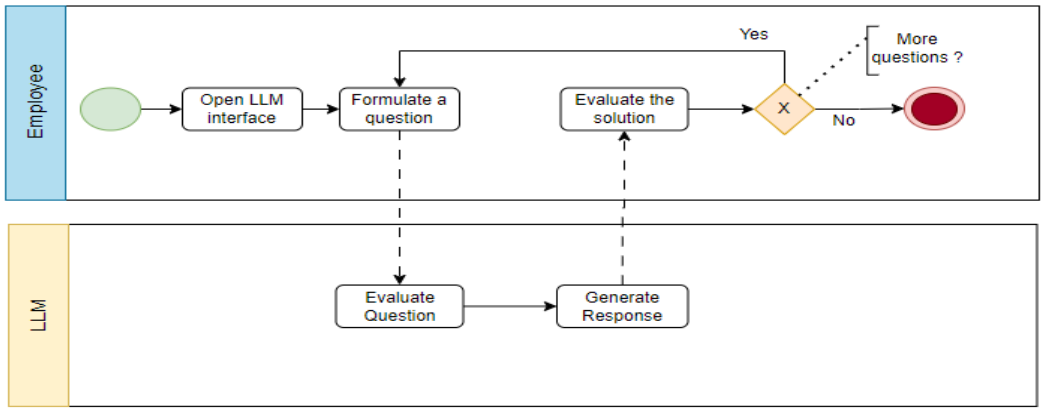

Figure 14. To Be process

As observed in the figure14:

- **First**, the employee opens the LLM interface, and instead of opening a ticket in Matrix42 platforms, the employee formulates a question directly into the LLM interface.

- **Secondly,** once the employee submits the question, the LLM comes into action. The model is trained on a vast and diverse dataset, enabling it to understand and interpret queries in real-time.

- **Thirdly**, after evaluating the question from the employee, the LLM delivers a response directly to the employee. The response time is almost instantaneous, which is a key advantage of using AI in the support process.

- **Fourthly**, the employee analyzes the answer from the LLM. If the answer responds to the question, the process is finish. If the answer does not fulfill the need of the employee, he asks a new question.

3.3.1 Functional Requirements

The functional requirements for the employee:

- The user interface should be intuitive and user-friendly, enabling smooth interactions for submitting queries and receiving responses.

- The system must include functionalities for creating, deleting, and initiating conversations.

- A toolbar library should be incorporated to keep a record of all past threads and conversations for the user.

3.3.2 Non-Functional requirements

In addition to the functional requirements, non-functional requirements were defined.

1. Security:
    a. The system must adhere to applicable data protection regulations for both the region and industry. This involves safeguarding the LLM host from cyber threats through strong encryption of data in transit and at rest. Furthermore, the LLM API should be designed to prevent unauthorized access and mitigate the risk of data breaches.

2. Accessibility:
    a. The system's design must prioritize accessibility, ensuring it is usable by a diverse range of users.

3. Performance:
    a. To achieve response speeds, the system could transmit each typed word to the backend in real-time. This allows queries to be pre-searched by a search engine and forwarded to the LLM before the user presses the enter or send button.

4. Scalability:
    a. The system must be scalable to efficiently handle the growing volume of user data.

5. Reliability:
    a. Availability: The system should be available 99.9% of the time, ensuring minimal downtime.
    b. Error handling: The system should gracefully handle errors and provide meaningful error message to users
    c. Data integrity: The system must ensure data consistency and integrity

4 Development

This section presents the development flowchart, which outlines the needed steps to perform the automated solution and explains in more detail the full steps to the development.

4.1 Flowchart

In this section the main steps demonstrated in Figure 15 will be explored, ranging from data extraction and cleaning to the generation of LLM responses.

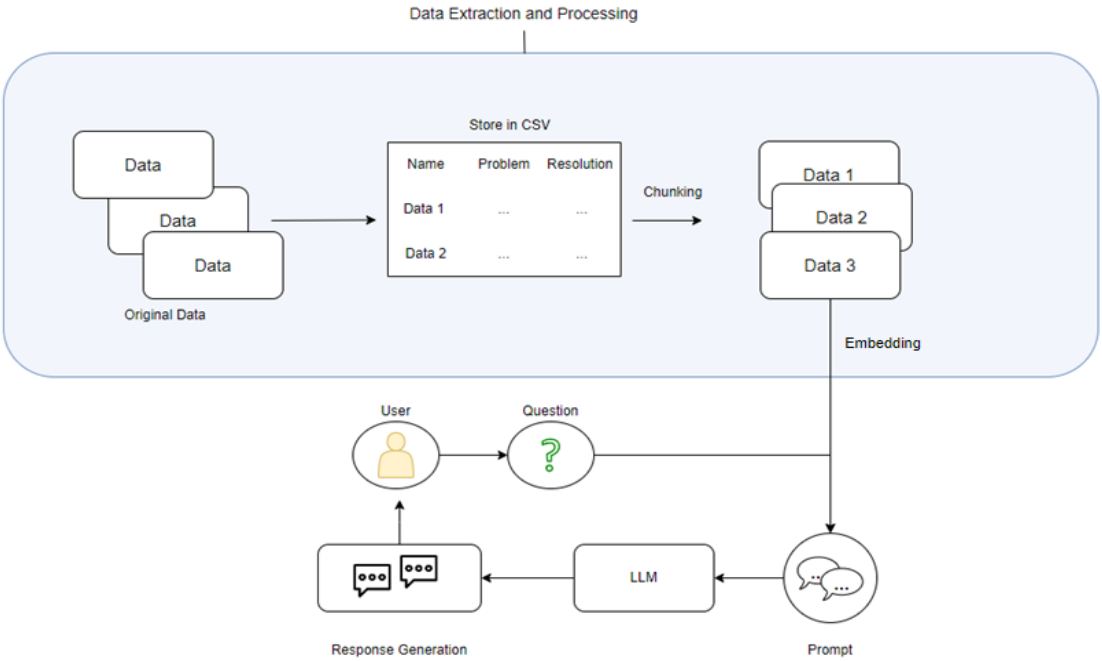

Figure 15. Development flowchart

Figure 15 illustrates the flowchart of the project, form data extraction and cleaning to generating LLM responses:

1. **Data Collection**: Collect the necessary information.

2. **Data Extraction and procession**: Extract relevant information from the original data and store it in a CSV file, structured into fields like "Name," "Problem," and "Resolution".

3. **Chunking:** Split the processed data into manageable pieces or "chunks" for easier handling.

4. **Embedding**: Convert the chunked data into embeddings, which are numerical representations that can be processed by machine learning models.

5. **User Query**: The user asks a question or provides an input query.

6. **Prompt:** The user's question is combined with the embeddings from the processed data to create a prompt. This prompt is tailored for the LLM to understand and process the query effectively.

7. **Response generation**: The LLM processes the prompt to generate a response based on the embedding and the user's query.

8. **Delivery**: The generated response is provided to the user.

The next sections will explain in more detail the flowchart and the steps involved.

4.1.1 Data Extraction and processing

The initial phase is crucial as it lays the foundation for all subsequent steps. The quality of data extraction and processing simplifies the performance of the LLM.

First, from Matrix42 application (original Data), five hundred tickets were extracted into a CSV format.

Table 8 shows a part of the extracted excel:

Table 8. CSV Data Extraction

Ticket ID	Full Description	Problem Category	Resolution
002	“Good morning, my computer is very slow...”	Performance issue	“Advised to clear cache..”
003	“Cannot connect to the internet...”	Connectivity issue	“Please make sure you are connected to...”

Second, the method chunking was applied to reduce the texts, making the LLM able to process the information, without losing important information and to improve the processing efficiency.

To apply the chunking method the following steps are necessary:

1. Import libraries: Pandas and Json libraries were imported for data manipulation.
2. Define JSON file path: Definition of the JSON file path that needed to be split.
3. Define the Chunk Size: Each chunk must be defined.
4. Save the Chung into a new JSON file.

Table 9 shows an example of chunking method:

Table 9. Chunking Method Example

The computer is slow, and the network connection drops frequently	
Performance issue	Connectivity

A ticket description like: "The computer is slow and the network connection drops frequently" could be chunked into two segments, one for the performance issue and another for connectivity.

Then, the models were chosen to apply in the incorporation stage, which transforms the data into geometric representations, allowing the LLM to "understand" the context, capturing the semantic relationships present in the information contained in the Comma-Separated Values file (CSV). The embedding algorithms that were chosen during the project were text embedding-ada 002 and Google's textembedding-gecko@001, as they are those that can generate adequate results.

4.1.2 Prompt development

One of the crucial aspects when interacting with LLM to obtain results in a specific area is the creation of prompts in a way that can guide the model to perform more accurately. This process involves the creation of templates, which are specific instructions on how the model should behave to perform a task. In this project, they are responsible for the activity of answering employees' questions in the field of company environment.

To obtain a more suitable result, it is necessary to define a context so that the model has enough data to perform a query and answer the employer's questions. Therefore, the following query was created:

> You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racists, sexists, toxic or illegal content. Please ensure that your responses are socially unbiased and positive in nature. If a question does not make any sense or is not factually coherent, explain instead of answering something not wrong. If you don't know the answer to a question, please don't share false information. The user can ask any information in this list ["Full Description", "Problem Category", "Resolution"] or using similar words.
>
> Answer the correct information to the question below as if you were a member of a support team:
>
> Therefore, based on the context below, answer the user's question:
>
> Context:
> {context}
>
> Question:
> {input}
>
> Response:

Figure 16. Prompt

4.2 Selected Models

Several models were used during the realization of this project, with the objective of finding the most suitable model. The models used were available on the HuggingFace platform Llama-2-70b-chat-hf, Falcon-7b-instructand and GPT-3, due to its popularity and proven effectiveness when performing response generation tasks. And for the vector embedding models the used models were: OpenAI's Text Embedding-ada 002 and Google's VertexAI's textembedding-gecko@001.

4.3 Response Generation

In this phase all procedures conducted in the previous stages will be applied with the help of the LangChain Library, which acts as an efficient connector between the context, the created prompt, the question asked by the user and the selected models.

The same parameter values were introduced for each selected model: temperature, max_new_tokens, repetition_penalty and top_p.

- Temperature: Controls the randomness of the results generated by the model. A lower value means the model is less likely to generate unexpected or unusual words. A higher value means that the generated text can be more creative, potentially leading to more incohesive responses.

- Max_New_Tokens: Controls the maximum number of responses generated by the model in terms of tokens. By limiting the number of tokens, it prevents models from generating exceptionally long responses.

- Repetition_Penalty: Controls the model's tendency to repeat words or phrases in its responses.

- Top_p: Controls the probability of choosing the words at a given stage of text generation.

After comparing the models' responses with the original text, the average values for each model were obtained based on the embedding algorithm that was applied. Thus, from Tables 11 and 12, it is possible to compare the performance of each model used during the project.

To show some of the results obtained, the following tables 11 and 12 will be created, each containing a question that will be applied to the models studied.

Table 11 shows the results obtained using the embedding model Text-Embedding-Ada-002. In turn, table 12 shows the results obtained through the embedding model textembedding-gecko@001. This way, it is possible to compare and evaluate the responses generated by the models in different scenarios, enriching analysis and understanding of their capabilities.

Table 10. Example of a Database with Questions and Context using embedding OpenAi

Embedding – Text-Embedding-Ada-002	
Question	What can I do when my computer is slow?
Falcon	"First, ensure that the system is managing the page file size automatically, as this can help improve your PC's performance…"
Llama-2-70b-chat-hf	"If your computer is running slowly, there are several steps you can take to try to speed it up. Here's a general guide: Restart Your Computer..."
GPT-3	"To speed up a slow computer, you can: 1. Close unused programs and tabs. 2. Restart…"

Table 11. Example of responses generated by Models using embedding HuggingFace

Embedding – textembedding-gecko@001.	
Question	What can I do when my computer is slow?
Falcon	"To improve the computer performance, please ensure the system is managing the file size in an autonomous way…"
Llama-2-70b-chat-hf	"If your computer is running slow, here are several steps you can take to improve its performance: Restart the computer, because…"
GPT-3	"A common issue many people face. Here's a step-by-step guide to help you troubleshoot and speed up your slow computer: Basic Checks: Restart the computer…"

4.3.1 Results and discussion

Based on the results obtained by the LLM models chosen in section, the BLEU Score metric was applied.

Therefore, a table named TableOriginalData was created as shown in Table 12, in which a small portion of the original data was selected directly from the source, consisting of 50 specific questions related to a particular topic. The purpose of this table is to allow a comparison between these texts obtained from the original source and the responses generated by the models. This comparison aims to evaluate how accurately the generated responses match the original text, providing a measure of the model's performance in generating accurate and relevant answers.

Next, based on Table 13, a performance evaluation was conducted for each of the selected models, generating BLEU Scores for the 50 questions.

Table 12. Example of table TableOriginalData

Original Data	
Question	**Answer**
How can I reset my SAP access password?	To reset your password, go to the Matrix42, click on ticket "Reset my password SAP," and follow the instructions to create a new one. If you encounter any issues, contact the IT support team for assistance.
How do I request reimbursement for expenses?	To request reimbursement, fill out the expense form in the finance system, attach the necessary receipts, and submit it for approval. Requests are typically processed within 7 to 10 business days.
How do I report a phishing email?	If you receive a suspicious email, do not click on any links or download attachments. Forward the email to the following email: emailsecurity@company.com for future analysis. If you accidentally clicked on a link or provided information, inform IT immediately.
How do I update my personal information in the HR system?	To update your personal information, log into the HR portal, navigate to the "Personal Information" section, and edit the necessary fields. Make sure to save your changes. If you are unable to make certain changes, contact HR for assistance.

Table 13. Result of models using OpenAI embedding

Embedding - Text-Embedding-Ada-002	
Models	**BLEU Score**
Falcon	0.15443
Llama-2-70b-chat-hf	0.08438
GPT-3	0.18672

Table 14. Result of models using HuggingFace embedding

Embedding - textembedding-gecko@001.	
Models	**BLEU Score**
Falcon	0.08213
Llama-2-70b-chat-hf	0.09164
GPT-3	0.009532

By leveraging the results from Tables 14 and 15, is possible to assess how closely the generated responses match the reference texts found in Table 13. This analysis is vital for understanding the quality and accuracy of the answers produced by the models. It allows to identify which models and embedding techniques are most effective for this project. In the context of this project, the primary goal is to determine the best models to respond to the most frequently asked questions by the company's employees.

Understanding the degree of alignment between the generated answers and the reference texts is essential, as it directly impacts the reliability and usefulness of the models in a real-world setting. The insights gained from this analysis will not only guide the selection of the most appropriate models but also inform potential improvements to the embedding strategies used. Ultimately, this evaluation process is critical to ensuring that the models deployed can provide accurate, relevant, and helpful responses, thereby enhancing the overall efficiency and satisfaction of the employees who rely on this system for their inquiries.

As observed in table 13 and 14, the OpenAI's embedding outperformed the textembedding-gecko@001 embedding. This performance gap was particularly noticeable across multiple questions, where the context generated by OpenAI's embedding produced more accurate and coherent responses, leading to higher BLEU Scores. The enhanced performance of OpenAI's embedding underscores its ability to generate responses that more closely align with the reference texts, making it a more effective choice for this task.

One potential reason for the inferior performance of the textembedding-gecko@001. embedding compared to the OpenAI embedding could be attributed to differences in their dimensionality. Specifically, the textembedding-gecko@001 embedding operates with 768 dimensions, while the OpenAI embedding utilizes 1536 dimensions [40]. This difference in dimensionality may account for the observed performance variations, as the higher-dimensional OpenAI embedding is likely to capture more intricate and detailed contextual information, resulting in more accurate and coherent responses.

The term "dimension" denotes the size of the vector used to represent numerical values in embeddings. Typically, embeddings with higher dimensions offer greater representational capacity, allowing them to capture more intricate and complex contexts [40]. However, this increased capacity comes with higher computational demands. In contrast, embeddings with lower dimensions are more resource-efficient but may provide less precise results.

As outlined in Section 5.3, the differences observed in the BLEU Scores can be explained by the specific function of the metric. The BLEU Score measures the overlap of words or phrases between the model-generated responses and a reference set of original texts (Table 12). Essentially, these metric measures precision by comparing how well the words in the model's outputs align with those in the reference texts.

Additionally, from tables 13 and 14, graphics were also created, for a better visualization of the results. These graphs are shown in Figures 21 and 22, which illustrate the performance of the model’s side by side, corresponding to the two embeddings used in the project. These graphs enabled the evaluation of performance using BLEU Score metrics, offering a comprehensive overview of how effectively the models performed across different text contexts.

Consequently, this analysis emphasizes the critical role of choosing the appropriate embedding to achieve accurate and relevant responses from language models. It illustrates how embedding selection impacts the model's performance in meeting user requirements and highlights the inherent complexity involved in optimizing LLMs for effective question-and-answer tasks. This understanding is crucial for customizing models to deliver high-quality results in diverse application scenarios.

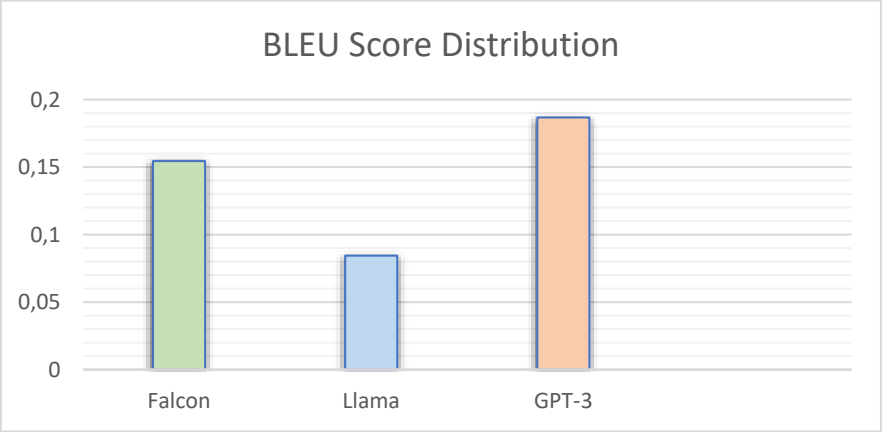

Figure 17. BLEU Score – OpenAI

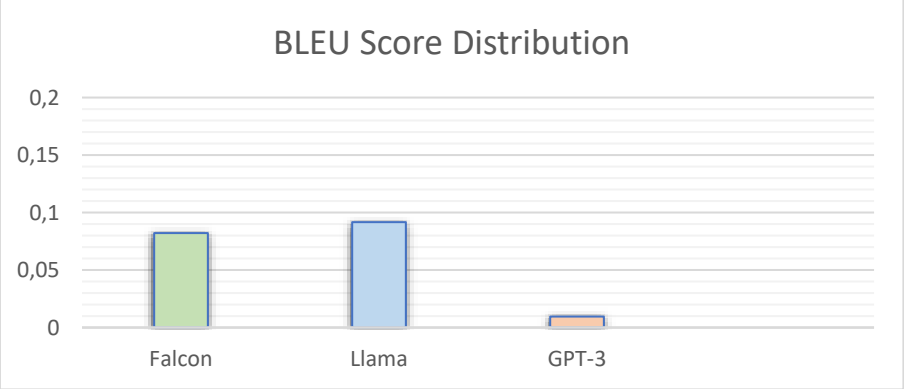

Figure 18- BLEU Score – HuggingFace

Figure 21 and 22 shows two bar charts comparing the BLEU scores of three models: Falcon, Llama, and GPT-3. In image 21 with OpenAI, GPT-3 has the highest score, followed by Falcon, with Llama scoring the lowest. In contrast, figure 22 shows Llama slightly outperforming Falcon, while GPT-3 has a significantly lower score.

5 Ethical considerations

When undertaking a project involving development and implementation of advanced technologies such as LLMs, it is crucial to address ethical considerations through the entire process. Here are some key ethical considerations to my project:

1. **Data Privacy and security:**
   Is important to ensure that employees data, including any information collected or processes by the LLM, is handled securely and in compliance with privacy regulation.

   a) Secure handling: Efficient data management begins with the secure handling of employee information. The LLM, being a repository of interactions and queries, must adopt encryption measures and access controls to prevent unauthorized access or data breaches. Employing industry-standard security protocols ensures the confidentiality and integrity of the stored data.

   b) Compliance with Privacy Regulations: Adherence to privacy regulations is non-negotiable. The LLM must align with applicable laws and standards, such as GDPR, or other regional data protections acts. This involves transparent communication with users regarding data collection, processing purposes, and obtaining explicit consent where required. Regular audits and assessments are essential to verify ongoing compliance.

2. **Transparency:**
   The employee must know that he is interacting with the LLM, not a human.

   a) Clear identification: To maintain transparency, it is crucial that the LLM is clearly identified as such from the outset. This can be achieved through an introductory message or visual clue that informs employees that they are engaging with an automated system.

   b) Limitations acknowledgment: An ethical LLM operation involves acknowledging the limitations of the technology. Users/employees should be informed about the specific capabilities and boundaries of the agent, making it clear that there are areas where human intervention may be necessary.

3. **Accessibility:**
   Ensure that the LLM is accessible to all users. Incorporated features that enhance accessibility must be considered.

a) In the ethical deployment of an LLM, primordial importance is placed on accessibility to ensure that every user, regardless of their abilities or preferences, can seamlessly engage with the system. Incorporating features that increase accessibility is not only a regulatory imperative but a commitment to creating an inclusive digital environment.

b) Multi-language support: In an increasingly globalized world, language diversity is a key aspect of accessibility. The LLM should be equipped with robust multi-language support, enabling users to interact in their preferred language. This ensure that language barriers do not stop effective communication.

4. **Employee Impact:**
   The impact of the LLM on employees and on the organization must be considered, especially those involved in support roles. Ensure that the introduction of automation is helping the employees and the support team.

   a) Efficiency gain: Evaluate whether the support team achieves efficiency gains with the introduction of the LLM. An effective LLM should speed up problem resolution, reducing response times and improving overall support efficiency.

   b) Resource allocation: Evaluate how the support team reallocates resources in response to LLM integration. This could involve assigning team members to more complex issues that require human intervention while allowing the LLM to handle routine queries.

   c) Employees feedback: Solicit feedback from the support team and from the other employees to evaluate their perspective regarding the LLM's impact.

Also, it is important to align the LLM with the General Data Protection Regulation (RDR. This RDR is an extensive data protection law implemented by the European Union (EU). This regulation is crafter to protect the privacy and personal data of individuals within the EU.

The RDR is applicable to all entities processing the personal data of EU residents, irrespective of their geographical location. It also extends its reach to companies beyond the EU that provide goods or services to individuals within the RU. The regulation outlines the key principles and obligations that organisations must observe while managing personal data. According to them, the processing of personal data should adhere to principles of lawfulness, fairness, and transparency.

6 Conclusivos

In conclusion, this thesis has meticulously explored the development and implementation of an intelligent agent aimed at enhancing the technical support processes at SEG Automotive Portugal, Lda. The research underscores the significant advantages of integrating advanced technologies such as Artificial Intelligence (AI), Natural Language Processing (NLP), and Large Language Models (LLMs) into the helpdesk operations.

The intelligent agent developed in this project addresses the critical need for efficient and immediate responses to employee inquiries, therefore helping the workload on the support team and improving overall operational efficiency. By automating routine support tasks, the agent not only reduces response times but also ensures the standardization of information provided to employees.

The findings from Chapter 4 revealed that the OpenAI embedding outperformed the textembedding-gecko@001 embedding, particularly in generating more accurate and coherent responses. The BLEU Scores indicated that the responses generated by the models using OpenAI's embedding were closer to the reference texts, highlighting the importance of selecting the appropriate embedding technique for optimal performance. Among the models tested, GPT-3 achieved the highest BLEU Score with the OpenAI embedding, demonstrating its superior capability in understanding and generating natural language text.

Future Recommendations:

1. **Build an API:** Developing an API is crucial for enabling seamless interaction between users and the LLM. The API should be designed to handle various types of queries and provide real-time responses. Key features of the API should include:
    - **Scalability:** Ensure the API can handle many concurrent requests without performance degradation.
    - **Security:** Implement robust security measures to protect user data and prevent unauthorized access.
    - **Documentation:** Provide comprehensive documentation to help developers integrate the API with other systems and applications.
    - **Error Handling:** Include detailed error messages and logging to help diagnose and resolve issues quickly.

2. **Develop a User-Friendly App:** Creating a user-friendly application would enhance the accessibility and usability of the intelligent agent. This app should feature an intuitive interface that allows employees to easily interact with the LLM, submit queries, and receive responses. The app could also include features such as:
    - **Real-Time Notifications:** Notify users when their queries have been processed and responses are available.
    - **History of Interactions:** Maintain a record of previous interactions to help users track their queries and responses.
    - **Feedback Mechanism:** Allow users to provide feedback on the responses received, which can be used to further improve the LLM's performance.
    - **Multilingual Support:** Ensure the app supports multiple languages to cater to a diverse user base.
3. **Enhance Training and Testing:** Continuously improve the training and testing processes for the LLM to ensure it remains effective and accurate. This could involve:
    - **Regular Updates:** Periodically update the training data to include new types of queries and issues that users encounter.
    - **Performance Monitoring:** Implement monitoring tools to track the performance of the LLM and identify areas for improvement.
    - **User Testing:** Conduct regular user testing sessions to gather feedback and make necessary adjustments to the LLM and its interface.
4. **Integrate with Other Systems:** Explore opportunities to integrate the intelligent agent with other enterprise systems used within the organization. This could include:
    - **HR Systems:** Integrate with HR systems to provide support for employee-related queries.
    - **IT Systems:** Connect with other IT systems to streamline technical support processes.
    - **Knowledge Bases:** Link the intelligent agent to existing knowledge bases to provide more comprehensive and accurate responses.

5. **Address Ethical and Legal Considerations:** Ensure that the implementation of the intelligent agent adheres to ethical and legal standards. This could involve:
    - **Data Privacy:** Implement measures to protect user data and comply with data privacy regulations.
    - **Bias Mitigation:** Regularly review and update the LLM to mitigate any biases in its responses.
    - **Transparency:** Maintain transparency in how the LLM operates and how user data is used.

In summary, this thesis has demonstrated the transformative potential of AI-driven solutions in technical support. The intelligent agent developed in this project not only enhances operational efficiency and employee satisfaction but also sets the stage for future innovations in the field of technical support. The successful implementation of this agent at SEG Automotive Portugal, Lda serves as a compelling case study for other organizations seeking to leverage AI to improve their support processes.

7 References

[1] I. Goodfellow, Y. Bengio e A. Courville, Deep Learning, MIT Press, November 18, 2016, p. 800.

[2] K. Garda, S. Nagpal, . D. Das, M. Dey e A. Mondal, "Chatbot: An automated conversation system for the educational domain," em *2018 IEEE International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 15-17 Nov. 2018.

[3] "SEG Automotive," SEG Automtoive, [Online]. Available: https://www.seg-automotive.com/. [Acedido em 20 10 2023].

[4] P. Ken, T. Tuure, R. Marcus e C. S, "A design science research methodology for information systems research," *Journal of Management Information Systems,* vol. 24, nº 3, p. 45, 2008.

[5] B. Kitchenham, T. Dybå e M. Jørgensen, "Evidence-based Software Engineering," em *Proceedings of the 26th International Conference on Software Engineering*, Washington DC, 2004.

[6] S. M, "Simplilearn," 7 Nov 2023. [Online]. Available: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-vs-machine-learning-vs-deep-learning. [Acedido em 16 11 2023].

[7] S. Russell, P. Norving e E. David, Artificial Intelligence: A Modern Approach, P. Hall, Ed., PEARSON EDUCATION LIMITED, 2010, p. 1132.

[8] H. Khan, “Types of AI | Different Types of Artificial Intelligence Systems fossguru.com/types-of-ai-different-types-of-artificial-intelligence-systems,” *ResearchGate,* vol. 9, p. 13, 2021.

[9] T. M. Mitchell, “Machine Learning,” em *Machine Learning Tom M. Mitchell*, USA, McGraw-Hill Science/Engineering/Math; (March 1, 1997), 1997, pp. 17-37.

[10] F. Rahioui, M. Ali Tahri Jouti, M. el Ghzaoui e P. Kumar Malik, “Machine Learning for Data Flow Processing in Learning Process,” em *IEEE, 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), 357-360*, 2023.

[11] N. a. G. T. Amruthnath, “A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance,” em *IEEE, 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 355-361*, 2018.

[12] R. S. S. a. A. G. Barto, Reinforcement learning: An introduction, Cambridge, Massachusetts and London, England: The MIT Press, ISBN: 978-0-262-19398-6, 2018.

[13] M. C. a. S. J. D. Pádraig Cunningham, “Supervised Learning,” em *Machine Learning Techniques for Multimedia: Case Studieson Organization and Retrieval*, Berlin, Heidelberg, Matthieu Cord and Pádraig Cunningham, 2008, pp. 21-49.

[14] J. Ebner, “Regression vs classification, explained,” July 2021. [Online]. Available: https://www.sharpsightlabs.com/blog/regression-vs-classification/. [Acedido em 20 11 2023].

[15] N. Hordri e S. Yuhaniz, “Deep Learning and Its Applications: A Review,” em *Conference: Postgraduate Annual Research on Informatics Seminar 2016At: Universiti Teknologi Malaysia, Kuala Lumpur*, Malaysia, 2016/10/26.

[16] F. Bre, J. Gimenez e V. Fachinotti, “Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks,” em *Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks, 158*, 2017.

[17] Hardegen, C. a. Pfülb, B. a. Rieger, S. a. Gepperth, A. a. Reißmann e Sven, “Flow-based Throughput Prediction using Deep Learning and Real-World Network Traffic,” em *2019 15th International Conference on Network and Service Management (CNSM)*, Halifax, NS, Canada, 2019.

[18] A. G. a. J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM network and other neural network architectures,” *Neural Networks,* vol. 18, pp. 602-610, 2005.

[19] K. Moaiad, “Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application,” *International Journal of Advances in Soft Computing and its Applications,* vol. 13, pp. 145-168, 2021.

[20] B. Billal, A. Fonseca e F. Sadat, “Efficient natural language pre-processing for analyzing large data sets,” em *2016 IEEE International Conference on Big Data (Big Data), 05-08 December 2016*, Washington, DC, USA, 2016.

[21] F. Gobet e P. C. R. Lane, “Chunking mechanisms and learning,” *Encyclopedia of the sciences of learning,* pp. 541-544, 2012.

[22] W. Qader, M. Ameen, Musa e B. Ahmed, An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges, 2019 International Engineering Conference (IEC), DOI:10.1109/IEC47844.2019.8950616, 2019, pp. 200-204.

[23] M. Azam Khan Raiaan, M. addam Hossain, K. Fatema e N. Mohammad Fahad, “A REVIEW ON LARGE LANGUAGE MODELS: ARCHITECTURES, APPLICATIONS, TAXONOMIES, OPEN ISSUES AND CHALLENGES,” *IEEE Access,* vol. 12, pp. 26839-26874, September 2023.

[24] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu e R. McHardy, “Challenges and applications of large language models,” *ArXiv,* vol. abs/2307.10169, 2023.

[25] U. Bezirhan e M. von Davier, “Automated reading passage generation with OpenAI's large language model,” *Computers and Education: Artificial Intelligence,* vol. 5, p. 100161, 2023.

[26] G. Rani, J. Singh e A. Khanna, “Comparative Analysis of Generative AI Models,” em *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)}*, Faridabad, India, 2023.

[27] A. Radford, K. Karasimhan, T. Salimans e I. Sutskever, 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)}, 2023, pp. 699-704}.

[28] J. Zhang, J. Huang, S. Jin e S. Lu, “Vision-language models for vision tasks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 46, pp. 5625-5644, 2024.

[29] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee e L. Zettlemoyer, “Deep contextualised word representations,” *ArXiv,* vol. abs/1802.05365, 2018.

[30] J. Wei, “ELMo: Why it’s one of the biggest advancements in NLP,” 2021. [Online]. Available: https://towardsdatascience.com/. [Acedido em 10 1 2024].

[31] S. Wang, C. Jiang e W. Zhou, “A survey of word embeddings based on deep learning,” *Computing,* vol. 102, 2020.

[32] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei e J. Launay, “The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only,” 2023.

[33] T. Brown, dvances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 1877-1901.

[34] M. AI, “Introducing LLaMA: A foundational, 65-billion-parameter large language model,” 2023.

[35] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell e ..., “Language Models are Few-Shot Learners,” em *Advances in Neural Information Processing Systems*, vol. 33, H. L. a. M. R. a. R. H. a. M. B. a. H. Lin, Ed., Curran Associates, Inc., 2020, pp. 1877-1901.

[36] . F. F. Xu, Z. Jiang, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan e G. Neubig, “Active retrieval augmented generation,” 22 Oct 2023.

[37] C. Delangue, J. Chaumond e T. Wolf, “Hugging Face - The AI community building the future,” 2016. [Online]. Available: https://huggingface.co/. [Acedido em 2 04 2024].

[38] “LangChain,” [Online]. Available: https://www.langchain.com/langchain. [Acedido em 1 5 2024].

[39] K. Papineni, S. Roukos, T. Ward e Jing ZhuWei, “BLEU: a Method for Automatic Evaluation of Machine Translation,” 2002.

[40] N. Reimers, L. Magne, N. Tazi e N. Muennighoff, “Massive Text Embedding Benchmark,” em *Association for Computational Linguistics*, Dubrovnik, Croatia, 2023.

[41] R. S. Wallace, “The Anatomy of A.L.I.C.E,” *in Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer,* p. 181–210., 2009.

[42] D. B. E. R. B. J. da S. Oliveira, “IBM Watson Application as FAQ Assistant about Moodle,” em *IEEE Frontiers in Education* , FIE, 2019.

[43] Gartner, ““Gartner Customer 360 Summit 2011: CRM Strategies and Technologies to,” 2011. [Online]. Available: https://www.gartner.com/imagesrv/summits/docs/na/customer-360/C360_2011_brochure_FINAL.pdf. [Acedido em 20 11 2023].

[44] Facebook, “Messaging with Businesses is Changing how People Shop Online,” [Online]. Available: https://www.facebook.com/business/news/insights/3-ways-messaging-is-transforming-the-path-to-purchase. [Acedido em 27 11 2023].

[45] J. F. Allen, em *Natural Language Processing*, GBR, Encyclopedia of Computer Science, 2003, pp. 1200-1230.

[46] I. R. e. al, “An empirical study of the naive Bayes classifier,” *IJCAI,* vol. 3, pp. 41-46, 2001.

[47] Z. C. a. X. Z. Jia Wu, “Self-adaptive probability estimation for Naive Bayes classification,” em *International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1-8*, 2013.

[48] J. R. Quinlan, em *“Induction of decision trees*, Machine learning 1.1, 1986, pp. 80-108.

[49] E. M. L. Adamopoulou, An Overview of Chatbot Technology, ResearchGate, 2020/05/29.

[50] Matrix42, “matrix42,” Matrix42, 2023. [Online]. Available: https://www.matrix42.com/en/. [Acedido em 20 11 2023].

[51] S. B. Kotsiantis, “Decision trees,” em *Decision trees: A recent overview*, Artificial Intelligence Review, 2013, pp. 250-290.

[52] BANU, SHAZIYA, PATIL e S. DEVI, “An Intelligent Web App Chatbot, 309-315,” em *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, Bengaluru, India, 2020.

[53] J. Weizenbaum, ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine, New York, NY, USA: Association for Computing Machinery, 36–45, 1966, p. vol. 9 36–45.

[54] G. V. Robertas Damaševičius, Information and Software Technologies, Vilnius, Lithuania, : 25th International Conference, ICIST 2019, October 10–12, 2019, pp. 4-7.

[55] H. Ho, P.-N. Cuong, L. N. H. Nam, N. Ho e e. D. Thang, Intelligent Assistants in Higher-Education Environments: The FIT-EBot, a Chatbot for Administrative and Learning Support, Conference: the Ninth International Symposium, 2018/12/06, pp. 62-80.

[56] S. Faleel e R. Nawarathna, “Artificial Intelligence,” In book: Artificial Intelligence. 10.1007/978-981-13-9129-3_14, July 2019, pp. 187-199.

[57] Godse, N. A. a. Deodhar, S. a. Raut, S. a. Jagdale e Pranjali}, “Implementation of Chatbot for ITSM Application Using IBM Watson,” em *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 1-5*, Pune, India, 6-18 August 2018.

[58] rASA, “Rasa: Developer Documentation Portal,” 2023. [Online]. Available: https://rasa.com/docs/. [Acedido em 30 11 2023].

[59] C. Sahu e M. Banavar, “A novel distance-based algorithm for multi-user classification in keystroke dynamics,” em *IEEE, 2020 54th Asilomar Conference on Signals, Systems, and Computers, 63-67, 10.1109/IEEECONF51394.2020.9443407*, 2020.

[60] E. Terumi Rubel Schneider, J. Vitor Andrioli de Souza e J. Knafou, “BioBERTpt -A Portuguese Neural Language Model for Clinical Named Entity Recognition,” em *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, 2020.

[61] C. Bartneck, C. Lütge, A. Wagner e S. Welsh, “What is AI?,” 01 2021.

[62] M. Teles Fernandes, “VALUE ANALYSIS: Going into a further dimension,” *Engineering, Technology and Applied Science Research,* vol. 5, pp. 781-789, Jan 2015.

[63] S. Russel e P. Norvig, Artificial Intelligence: A Modern Approach,, United kingdom: PEARSON EDUCATION LIMITED, 2021.

[64] E. Davis, P. Norvig e S. Jonathan Russell, Artificial Intelligence: A Modern Approach, Prentice Hall, 2010.