Open University of Cyprus

Faculty of Pure and Applied Sciences

Postgraduate (Master's) Programme of Cognitive Systems

Master's Thesis

Implementation of a Conversational Agent for Customer Support

Stavros Leonidis

Supervisor
Isidoros Perikos

November 2021

Open University of Cyprus

Faculty of Pure and Applied Sciences

Postgraduate (Master's) Programme of Cognitive Systems

Master's Thesis

Implementation of a Conversational Agent for Customer Support

Stavros Leonidis

Supervisor
Isidoros Perikos

The present Postgraduate (Master's) Dissertation was submitted
in partial fulfilment of the requirements for the postgraduate degree
in Cognitive Systems
Faculty of Pure and Applied Sciences
of the Open University of Cyprus.

November 2021

This page was left intentionally blank

Abstract

The aim of this Master's Dissertation is to implement an AI-based framework that currently is used today by many retail companies. This framework which incorporates conversational agents, includes an interaction between a human and a machine. Conversational Agents aim not only to provide a smoother and faster customer service for a commercial enterprise but also to increase enterprise's business performance. At this Master's Dissertation, the technical features that are common characteristics to most agents are examined together with the explanation of the process which agents are based in order to respond to simple questions. The implementation of the agent has been performed via the open source software Rasa.

Περίληψη

Ο σκοπός αυτής της μεταπτυχιακής διατριβής σχετίζεται με την υλοποίηση ενός συστήματος που βασίζεται σε αρχές τεχνητής νοημοσύνης και χρησιμοποιείται σήμερα από πολλές εταιρείες λιανικής πώλησης. Αυτό το σύστημα που ενσωματώνει διαλογικούς πράκτορες, περιλαμβάνει μια επικοινωνία μεταξύ ανθρώπου και μηχανής. Ο κύριος σκοπός των διαλογικών πρακτόρων δεν είναι μόνο να παρέχουν μια ομαλή και γρήγορη εξυπηρέτηση πελατών σε ένα εμπορικό οργανισμό αλλά και να αυξήσουν την επιχειρηματική απόδοση του οργανισμού αυτού. Σε αυτή την μεταπτυχιακή διατριβή, τα τεχνικά χαρακτηριστικά που είναι κοινά στοιχεία στους περισσότερους πράκτορες αναλύονται καθώς εξηγείται και η διαδικασία με την οποία οι πράκτορες είναι σε θέση να απαντούν σε απλά ερωτήματα. Η δημιουργία του πράκτορα έχει υλοποιηθεί μέσω του λογισμικού ανοικτού - κώδικα Rasa.

Acknowledgments

I wish to express my deepest gratitude to my supervisor Dr. Isidoros Perikos who gave me the opportunity to start this Master's Dissertation on the area of conversational agents and for guiding me throughout this process. I also feel grateful to my parents, my brother, my dog and my friends, who stood by me and supported me at all times during the whole journey. This work would not have been possible without all of you.

Table of Contents

Chapter 1
Introduction

The use of computers and smartphones have become a reality for all people and are considered, by many people, as an integral part of our lives. It is a reality that every day users in order to fulfil either their personal needs or solve many of their problems, utilize such methods to exchange information via written or oral means of communication. The utilization of these devices can be exploited even by retail companies or online stores, in order to strengthen their position in the market and simultaneously offer better service for their customers. Therefore, companies, in the context of business competition, are progressively trying to develop/apply techniques and technologies in order to maintain and expand their customer base. The traditional way of providing services to customers through talking to a company representative or communicating online via email tends to be reduced in favor of new alternative schemes that utilize artificial intelligence and offer more immediate and faster customer service (Li, Chenzhuoer & Pan, Runjie & Xin, Huiyu & Deng, Zhiwen 2020: 2). Until recently, in cases where a potential customer wanted to order a specific product, or in cases where a client enquired for the progress of his/her order, the only way of communication with the company was either to send an email to the sales dept. or to call the customer service team and wait for a certain time, in order to be served. However, this process seems to be time consuming as the customer can often have to wait from several hours to several days to receive a response to his/her request, or in the case of a telephone conversation, to wait for a queue for his/her turn. It is a fact that the people working at the call centers do not always manage to serve all the clients/potential customers immediately, and sometimes customers' frustration / disappointment could be considered as potential outcomes of this process.

A solution to this problem is provided by conversational agents (CA), irrespective they are voicebots or chatbots. Through CA, companies utilize information technology and more specifically artificial intelligence and natural language processing / comprehension to serve all customers whenever they wish. These systems are a possible solution to the daily needs of consumers to obtain the information they want with precision and immediacy. They are able to provide targeted information and answer users' questions at any time of the day, as there is no need to interfere with human presence during their operation (Zumstein & Hundertmark 2017: 14). As a result, the company's reputation and profitability will rapidly be increased.

In the context of this Master's dissertation, an introduction to the wider area of artificial intelligence and natural language processing is provided, with a related literature review of what has been achieved so far. Following this introduction, this Master's Dissertation then discusses the role of the Online Customer Experience in the present time and how it has changed in recent years by incorporating the technologies mentioned earlier. In the next chapter, the principles of CA are introduced, starting with a definition and the basic characteristics that govern an agent. Important topics, such as how an agent can integrate cognitive characteristics and what information and linguistics technologies an agent could incorporate to function effectively and answer user's question are also discussed. Finally, a short discussion to the existing agents that are commercially available and are utilized by the largest companies for customer service is provided. For the practical part of this Master's Dissertation, the implementation of the conversational agent has been considered. The conversational agent is based on Rasa software, a widely used open source software for developing such agents. The implemented conversational agent focuses on the customer service in an electronics store and is trained to answer questions or resolve customer issues regarding the order or the progress of an existing order. Obviously, in order to build a complete agent which can be able to answer any customer question, a team of conversational designers need to be employed to fully train the agent. Therefore, the purpose of this Master's Dissertation is to present the “philosophy” of such an agent and its implementation techniques. In the last part of this thesis, an evaluation of the agent is made and relevant results are presented.

Chapter 2
Preliminaries

2.1 Artificial Intelligence

The term of Artificial Intelligence (AI) was initially introduced during 1950s, when Alan Turing published a paper which, for many scientists, initiated the birth of AI. The main idea in that paper was the famous "Turing Test" concept, which focused on measuring the intelligence level of machines and to decide if such machines could convince an evaluation panel that they are in fact human rather than machines (French 2000: 3). AI is an important area in computer science with the main target to design/implement computer systems that could operate in an intelligent way, imitating thus human-like actions. Any machine that performs a tasks in a similar way like a human, it is considered to include some levels of intelligence. AI frequently is referred to as "intelligent code" rather than as "AI", and such definition depends on the viewpoint of the respondent. Computer scientists often seem to use the term intelligent code while business-oriented people prefer the usage of the words artificial intelligence.

In order to understand the concept of AI, we need also to address the issue of what the purpose of AI is. The main goal here is to create machines/systems that can think and/or act like the way humans operate. Based on Tesler's theorem "Intelligence is whatever machines haven't done yet", the meaning is that what a machine can do, is not human intelligence but something else than intelligence. This is the so-called "AI effect", which means that when big advances have been occurred with AI, like for example when computers win chess games, critics consider it irrelevant to AI but just computation. However, what was considered as intelligence in the past, it is something normal today, thus the definition of intelligence can vary (Bezboruah & Bora 2020: 3). With AI, any business company for example can "utilise" the power of data, personalization and provide almost perfect services to customers that are available 24/7. The keyword to be emphasized here is, "with". Computers alone are not able of performing such actions alone. Human's interaction is vital in teaching the computers, analysing and implementing their outcomes to real life case studies. It is this specific engagement of humans and machines that could improve our business performance.

AI is considered as the machine's ability to recognize patterns, sounds, images and to learn over data. Some of these tasks, made likely by AI-based algorithms, include learning, problem solving, decision making and prediction by gathering big amounts of data and analysing them. Practically, it is a collection of techniques that enable computers to understand and interact with the world more naturally and responsively than in the past by using data. However, humans are still better in analysing emotional intelligence than computers utilising AI-based algorithms. In general, the AI algorithms of today have not changed too much since the 1990's (even the deep learning systems are based on existing learning schemes) but the current computational power is making changes rapidly. Today, computers are able to process larger amounts of data than before and the storing of larger amounts of data (i.e. big data) is now possible.

AI technologies have the real potential to complement and support specific tasks that are currently operated by humans today. However, if AI is provided with human-equivalent rights and general access to decision-making processes, this could lead to AI-based systems to take control and consequently question the necessity of having humans in such tasks. Obviously, this issue is direct related to ethics and transparency issues involved with the use of AI. On the other hand however, AI has the potential to transform our society and to create a new environment where such systems could supplement human contributions and individuals could spend more time to improve/engage with personal development and individual desires. As examples of such intervention, AI has been utilised to automate many aspects of our lives, like autonomous vehicles (Tesla), robots for home cleaning (Dyson), digital nurses that monitor the condition of a patient (Sensely's virtual assistant "Molly"), and voice-controlled speakers (Amazon Echo Dot) to manage home accessories (Mohsienuddin & Syed 2020: 5-7).

Although the trends in AI suggest that humans could be substituted not only in tasks that require "mechanical intelligence" (like call centre agents, waiters, taxi drivers) but also in tasks that require analytical and intuitive intelligence (like accountants, analysts, managers), there are still reservations that AI eventually could become a serious concern (i.e., replacing humans). It has been recommended that human consciousness may not be able to be simulated in rules, and machines can only be programmed to follow rules or even to autonomously create their own simple rules. This assumption means that it may be difficult to design/create an AI-based system that fully emulates human-like intelligence. A chatbot is one such example of AI that utilises Natural Language Processing (NLP) to communicate and interact with humans. NLP is a combination of techniques/algorithms that include the analysis of text or language on various levels. Through NLP-based systems, humans can interact with AI using natural language. In some cases, improved models, like the adaptive response model, have been used to understand what humans are saying and thus to provide adaptive responses. This type of models is considered to increase observed humanness and provide faster responses. Chatbots utilise the same kind of dialogue patterns as in SMS and messaging applications like Facebook Messenger. Since 2014, companies/developers have created chatbots inside these applications, which has eventually increased the accessibility of chatbots.

2.1.1 Important milestones in AI and computing

Artificial Intelligence is not a new application domain, and a lot of research has been conducted in this filed in the past 70 years. AI has gone through stages of stagnation and growth. The times of “stagnation” occurred from 1970-1980 and again from the beginning of 1990 to 2000. In those periods, not much funding was invested in this area. These milestones are historically important because they are the “enablers” for using AI-based solutions in various applications today. Most research papers consider the 1940’s to be the first milestones in computing. In 1943, McCulloch and Pitts proposed to build computers whose components resembled neurons. Eckert and Mauchly from Pennsylvania University together with their team developed the world’s first electronic programmable computer ENIAC (Electronic Numerical Integrator and Computer) for the purpose of conducting calculations for the first atomic bomb in 1945 (Brandao 2018: 3). Computational statistics and machine learning were “born” in 1950 when Turing proposed that if the computer can mislead the human to consider it is the other human, it has then artificial intelligence. John McCarthy, created the LISP programming language for AI in 1958, which became together with also Prolog a useful tool. Between 1960 and 1980, NLP, computer vision and robotics thrived while on the other side, the development of AI was frozen. The development of personal computer was an important milestone in the advancement of computing, because more people had more access to computers. Actually, in 1982 the Times magazine declared the first personal computer the “machine of the year”. After this milestone, CERN created the World Wide Web (WWW) in 1989 for internal purposes initially, and since then the era of information sharing and retrieval begins. The first smartphone was created by IBM in 1992, a movement that transformed the way we communicate/interact with others today. These milestones are some of the important ones in the history that have led us to the technological point we are currently.

At previous years, although we were witnesses of advances in the area of AI, the lack of capacity to store data and utilize it, were the biggest obstacles for further applicability of AI-based applications. However, currently, we have the capabilities to store massive amounts of data, transform new types of data, such as pictures and speech, into formats computers can process, so the evolution of new technologies have started flourish. Amazon, Google and Microsoft are among the biggest enterprises that offer various types of AI-based solutions such as speech to text, text to speech, image recognition and sentiment analysis.

2.2 Machine Learning

Machine learning (ML) is considered an area of AI that utilises data for learning tasks and categorize/process them with marginal human involvement, rather than being programmed to do a certain task, in other words it automates the process of learning on its own. Most of what we consider today as AI is actually machine learning systems. ML is an algorithmic area that combines concepts from statistics, computer science and many other disciplines to design algorithms that process data, make predictions, and help make decisions. Machine learning is practically a “family” of algorithms that are programmed to learn from given data and are progress gradually (i.e. learn) via a training learning process. An algorithm is practically a set of rules, a program that gives the computer instructions on how to perform a task. In machine learning the data is divided in to training data and testing data. The training dataset is fed to the model first to predict a certain outcome and after the end of this process, the test dataset evaluates how well the model has been trained. As illustrated in the following schematic, ML is divided into several subcategories: supervised learning, unsupervised learning and reinforcement learning (Çelik 2018: 3-4).

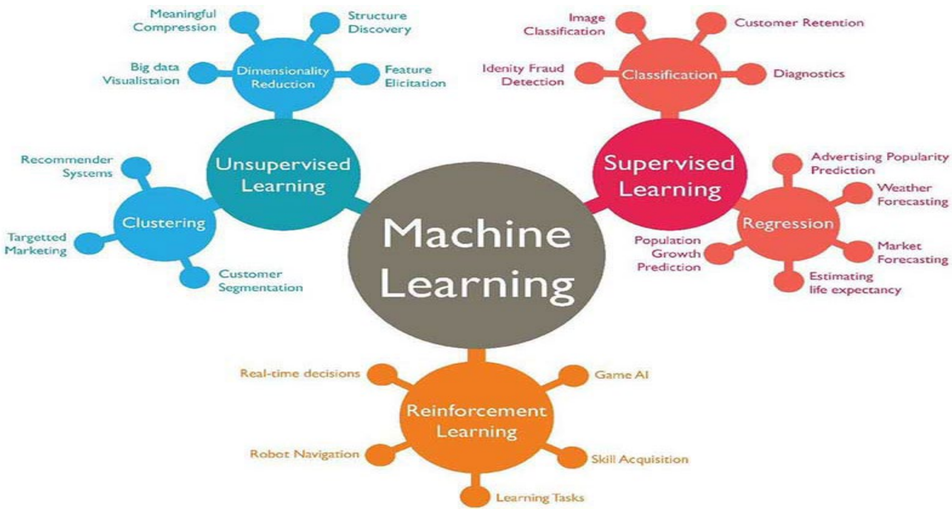


Figure 1: Types of learning methods

These different types of learning methods are used when training ML-based algorithms for a particular task.

- Supervised learning is used for training a model by giving it labelled training data that includes sets of training examples and labels and the known outcome of the task. Supervised learning can be used either for classification or regression applications. Classification examples may include speech recognition, image classification, identity fraud detection and measuring customer retention. A classic example of regression is the financial forecasting where a ML algorithm tries to predict upcoming values/indices of a specific financial feature.

- Unsupervised learning is important in providing different automated explanations at a faster pace with minimum interference of humans. In unsupervised learning the outcome of the problem is unknown and the model considers some assumptions based on the regularity and relation of the data given (i.e. similarity issues for example). Examples of unsupervised learning include clustering and dimensionality reduction methods. Dimensionality reduction schemes is a method of reducing the complexity of problems, especially those which include too many attributes (input variables). Clustering means to group text, images, audio, numerical or mixed data objects into clusters, in which the characteristics of these included objects resemble one another more closely within this cluster.

- Reinforcement learning is a different type of learning process and is based on previously defined parameters by which it can independently analyse and decide a particular action. In reinforcement learning, the machine is provided with a feedback on how well it is performing rather than giving the correct desired outcome, in order to minimize the risk and maximize benefits. Reinforcement learning is used in the area computer games, robotics and real-time decision systems.

2.3 Natural Language Processing

Natural Language Processing (NLP) is a very important component in AI domain. NLP is a division of computer science that allows computers to extract and/or generate meaning from a text that is understood by humans and is correct in grammatical terms. Humans, unlike machines, express themselves in different and sometimes in complex means. The way we communicate either verbally or written goes much deeper than what the actual meaning of a word is. There are limitless ways to express an idea. This is due to the fact that modern languages have so many different words and even the use of the same words may have a different meaning based on the context of use. Thus, an NLP system is desirable to handle such different scenarios and at the end of process to provide as an outcome the understanding of what was meant by the collection of words. According to SAS Institute of analytics, the tasks NLP is capable of, can be utilised to (Annika 2020: 18).

- Categorize content, create alerts and detect any duplicates that exist in a text.
- Discover patterns in the text that could be utilised for optimization and prediction, for example to personalise the content for a customer
- Sentiment analysis which can identify individual opinions from large amounts of text. This can be especially useful for the case of chatbots to identify for example, customer opinions about a specific product in online platforms.
- Convert voice commands into text and vice versa, written text into voice commands.
- To provide automated translation of text into another language.

Based on this list of tasks, NLP and text analytics can be combined together in business related applications, for example to identify patterns and clues in emails, written reports or customer feedback in different platforms. Google has utilised a lot of these capabilities available for users such as Gmail for filtering spam, Google keyboard Auto-correct, Auto-predict from Google search and machine translation from Google Translate. Therefore, NLP is the tool that allows machines to "understand" human language. This means, in practice, that humans do not need to know any programming languages to interact with the machines utilising NLP tools.

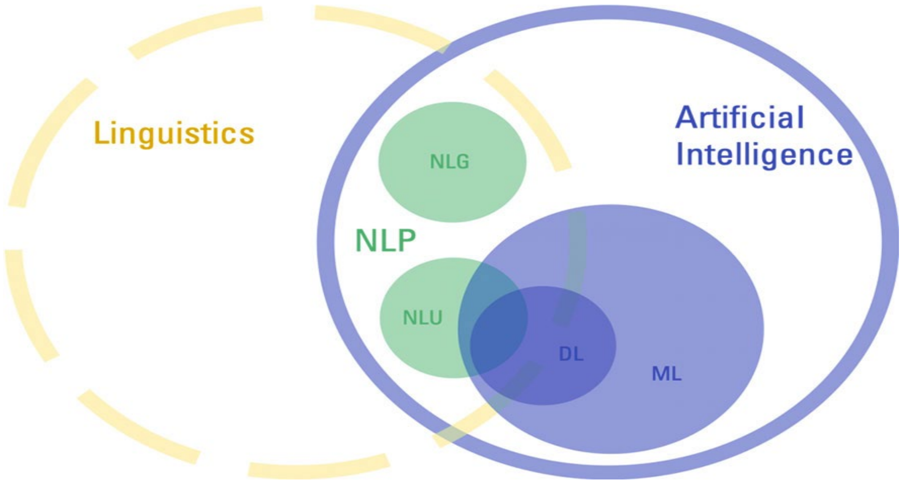


Figure 2: Natural Language Processing area

NLP framework is divided into two components, the natural language understanding (NLU) and the natural language generation (NLG). NLU component provides the machine the understanding of a natural language input. However, the adopted methodology of NLP is not to process the input word by word but it evaluates the sentence in the way humans do. The understanding of natural language involves three stages. Firstly, the words' meanings need to be defined. Then, the meaning of the sentence is determined as a whole by checking the order of the words and the related grammar. Finally, the meaning of the sentence is evaluated according to its context and the field of use. NLG is the second component of NLP and is responsible for generating the output. This is the mechanism that occurs after the NLU intervention, and as at this stage, it is clear for the machine what the user's intention is, a specific output is then generated consequently. NLG finalises the communication process by selecting the wording, checking the grammar and finally creating a word or a phrase as a response to the user's query. NLP is usually used interchangeably with AI and ML (Dale 2020: 4-7).

However, they are different from each other. NLP and ML are considered as parts of AI. AI can assist machines to solve complex important for humans' problems. NLP is the system that assists machines to understand the way humans communicate in both written and verbal language, while ML is a system which enables the machine to learn from its own observations and previous experiences. The above schematic illustrates these relations.

Chapter 3
Conversational Agents

3.1 Online Customer Experience

Currently, in our society, dominated by various media, the experience characteristic has become the main issue of businesses differentiation strategies. Based on this new "business environment", customers have shown preference to the concept of Customer Experience (CX) rather than to the traditional way of transaction-based customer relation scheme. There are various research studies addressing CX as a significant part in adding value either for customers or for the businesses. Klaus and Maklan (2013) tested a number of different hypotheses by exploring relations between a positive CX and possible outcomes like customer satisfaction and potential loyalty intentions. All their proposed hypotheses were confirmed. Gentile, Spiller, Noci (2007) have defined the concept of customer experience as a subjective response of a customer to his/her interaction with the specific product or the actual company that produces this product. Based on their study, CX includes a number of sensorial components such as senses stimulation like vision, hearing, and touch. In also includes emotional components like moods, feelings, as well as a cognitive component such as mental processes. Verhoef, Lemon, Parasuraman, Roggeveen, Tsiros, Schlesinger (2009: 2) further extended this definition with a more generic approach by splitting customer responses into cognitive, affective, emotional, social and physical parts. He has also emphasized the fact that customer experience includes all possible phases of the procuring process, like search, purchase, consumption and obviously after-sales.

In today's business environment, the concept of "online website" is gaining more and more significance. Hoffman and Novak (1996) proposed that in order for an online website to be effective in attracting its potential visitors, it must enabling a "state of flow". They defined this term as the state that occurs during navigation in the network and is characterised as a continuous sequence of responses facilitated by a machine-user interaction, which is assumed to be essentially enjoyable and is supplemented by a loss of self-consciousness. Visitors in these online websites can attain flow when they manage to have a balance between their skill level and possible challenges related to this online interaction. Flow generally increases user's investigative and participating behaviour. On the other hand, there is a possible risk of visitors being distracted. Visitors, may wish to concentrate on exploring the website rather than trying to find their required information. Thus, Hoffman and Novak, utilised this concept of flow as a fundamental component for their process model of network navigation in the hypermedia Computer-Mediated Environments. They have identified four Flow factors, all of them associated with user's cognitive responses (Hoffman and Novak 1996: 8)

- perceived congruence of skills and challenges,
- focused attention,
- interactivity,
- telepresence.

The first two components are important for the creation of the flow state to occur, while the remaining ones enhance it. Hoffman, Novak, Yung (2000), using results from various surveys related to a large web-based consumer sample, they managed to test thirteen individual hypotheses from the model introduced in their previous paper. The authors investigated relationships between hypotheses and consumer behaviour as well as related web usage. Based on their conclusions, the mood of fun and exploration drops, the longer visitors use the web. They claimed that the flow would be bigger for visitors utilising web for experimental purposes rather than for goal-oriented tasks. However, their claim was rejected in one of their later research work, where it was suggested that that experiencing flow was more dominant among goal-oriented users (Novak, Hoffman, Duhachek 2003).

Most of their hypotheses were confirmed through extensive validation processes. However, the increased focused attention did not increase the flow. Nevertheless, as this attention was associated with greater telepresence and time distortion, it was considered as an indirect factor. Among new findings from this research was that the interactive speed was proven to have a positive outcome on the flow, challenge was positively linked with focused attention and finally importance was beneficial to the skill level of the user. Rose, Clark, Samouel, Hair (2012) investigated CX in online shopping context. Their study, in addition to cognitive components, incorporates also affective ones. Further studies suggested additional variables such as playfulness and personal innovativeness, content/interface, novelty or perceived usefulness. Finally, Hoffman, Novak (2009) reviewed their model by creating a survey on other works in online customer experience field. They emphasized a diversity of measurements proposed in various papers. They found that single-dimensional measures tend to be simpler and easier to collect the data for. However, multi-dimensional measures can allow a more holistic definition of flow able to be tested for fitting in a structural model. Therefore, multiple measures of flow were recommended for use when it is possible.

3.1.1 Conversational Agents

Technological advances in the domain of business have revealed a growing interest in exploring Conversational Agents (CA). Estimates indicate that the application of such agents will be accountable for cost savings of $11.5 billion by 2023. These achievements originate from the scalable as well as cost-effective services provided by CA systems, focusing on the efficient use of technology and improving service quality and market competitiveness. Although such services increase the user engagement by offering customized flexibility, they are prone to certain limitations arising from the grammatical complexity or semantics of the conversation. A CA can be defined as an interface to users to interact with a computer, based on human terms. In the early ages of such interaction, users had to input specific, in terms of syntax, commands to the computer. CAs thus can be considered as more “social” compared to the traditional technological applications, from the viewpoint of the user who is involved in this interaction. Obviously, CAs are independent of the platform and they can be integrated into a number of various platforms, like Google Assistant, Alexa, Cortana, Facebook Messenger and many others. In general, CAs are grouped, based on their interaction mode, into two types, the chatbots and voicebots (Hussain, Sianaki, Ababneh, Nedal 2019: 2).

Chatbots: A chatbot is a computer program that interacts with users by simulating a human conversation to assist with their queries. Prior to the development of chatbots, users used to navigate through various graphical interfaces to search for what they were looking for. However, with the creation of chatbots, a new way of interacting with the software has been introduced. Many websites currently incorporate a chatbot which simplifies the process of searching. The user needs to use only the chatbot’s interface to make it understand what his/her query is. In general, there are two main types of chatbots, the AI-driven or NLP-based chatbots and the rule-based chatbots. The first type of chatbots makes use of an underlying AI mechanism, while the latter type is more of a hard-coded version (if-then statements).

- AI-Driven Chatbots: AI-driven chatbots are based on the idea of NLP since these chatbots incorporate NLP principles as their main mechanism together with ML algorithms. AI-driven chatbots are thus able to simulate a human conversation. They are efficient at recognising user's intention and provide relevant responses. As an example, in a user's respond like "Why not?" as an answer to a chatbot's suggestion for a product, the chatbot understands this answer as a confirmation instead of a question. In addition, AI-driven chatbots incorporate ML algorithms in order to strengthen their development over time, thus improving their predictive ability to provide useful answers.
- Rule-Based Chatbots: Rule-based chatbots are programmed for specific tasks and thus can provide replies to specific type of queries they have been hard-coded for. Obviously, these tasks can have different levels of complexity, addressing thus from simple to complicated queries. However, due to the nature of design, it is very difficult for rule-based chatbots to handle all the cases in a particular situation.

Voicebots: Voicebots are an alternative type of CA that utilise speech recognition technology to interpret natural language commands. Voicebots can be considered as voice assistants. Voice assistants are digital interfaces that emulate a human-to-human conversation flow. They use sound as a means of communication and remove the need for a graphical interface. Similarly to the case of AI-driven chatbots, voicebots are incorporate also AI and ML based algorithms in addition to voice recognition technology. Voice assistants include a dedicated software for listening to the sound around them. When a keyword is recognized by this software, a mechanism is activated to facilitate the voice assistant. Following the activation process, the user's voice is recorded and sent to a dedicated server for further analysis and interpretation. The server then provides the voice assistant with the necessary information it needs to be able to answer the user. Voice commands are getting more and more supported by various services. Internet-of-Things manufacturers are also adding compatibility for voice commands in their devices. Siri, Google Assistant, Alexa and Cortana are some of the most well-known voice assistants developed by the tech giants Apple, Google, Amazon and Microsoft respectively. They are integrated into smartphones and systems and provide a number of functions, including controlling home devices, playing music and answering questions. The assistants have integrated voice matching technology to identify the user and personalize requests.

- Apple Siri: Siri was developed by the Cognitive Assistant that Learns and Organizes (CALO) project which was launched in 2003 by the Defence Advanced Research Projects Agency (DARPA) as a digital military assistant. Although CALO project was developed initially for desktop computers, Siri was designed especially for mobile devices. A first version of Siri was available in early 2010, and was bought by Apple Inc. a few months later. In 2011, Siri became an integral part of the iOS operating system.
- Google Now & Google Assistant: Google offers two types of Intelligent Personal Assistants (IPA) - Google Now as part of the Google App, and Google Assistant. Google Now was announced in 2012 and its main purpose was to offer a natural language interface to the Google search engine. Similar to Siri, it can be activated by a voice command. Google Assistant was announced in 2016 and can be considered as an improved version of Google Now. It is not only available as a mobile app, but also integrated in a voice-activated speaker called Google Home.
- Amazon Alexa: With Alexa, Amazon also participated in the development of IPAs. It was first announced in 2014 as part of Amazon Echo, a speaker with integrated IPA.
- Microsoft Cortana: Cortana is an IPA developed by Microsoft and was released in 2014. It is integrated into recent versions of the Windows operating system and also available for the mobile platforms Android and iOS.

3.1.2 Chatbots to improve customer experience

A chatbot is considered as a computing system with a main task to simulate a conversation with a human over the internet. Chatbots are categorised into various types and these include product and customer service chatbots, informative chatbots and virtual personal assistants (Følstad, Skjuve, Brandtzaeg, 2019: 7-9) In marketing and sales sector, chatbots can support a company offering a more personalized service to their customers, based on customers' information extracted from internal and external databases. Such engagement with customers via chatbots is a common practice today, because it helps the humans working in customer service departments in their tasks. In addition, as chatbots can operate 24/7, this is considered as a valuable asset for the company to be able to support the customers even outside of office hours. Current chatbots are based on "question and answer trees" philosophy. They are able to classify and extract basic information before passing on to human operator a specific customer situation. Natural Language Processing (NLP) can be used to understand the background of customer inquiries and the actions will be determined by the tree structure. Current chatbots work more like as an internal search engine, which is able to direct customers' requests to the correct solution. Their usage in customer interfaces is the most common today. Through the time, chatbots are implementing to become more conversational and intelligent. From the early stages, when a very basic chatbot that is rule-based and replies in pre-determined words without any understanding of language to the latest trend, where the most cognitive bot which can solve problems by itself. By using such intelligent chatbots, a company can improve the customer experience in different ways, for example by providing 24/7 customer support or selling products without the involvement of human personnel.

Chatbots use AI while communicating with humans. The more the bot interacts with humans or each other, the more it learns to answer in a more intelligent way. Chatbots have been provided with prearranged answers to questions that it tries to answer and if it is unable to provide correct answers then the conversation can be transferred to a human operator. This way of operation is beneficial for the company, as it saves time of customer service representatives. Chatbots can also send personalized messages, videos, even offers to customers and can be programmed to interact in a natural way is appreciate by the customer. They can also be programmed to collect and analyse customer feedback in order to improve the quality of the service and even sometimes to offer promotional offers to customers for further purchases (Duijst 2017: 5). The most advanced chatbots are capable in analysing sentiments in text and pictures, a characteristic that can improve the customer experience service. Another advantage of chatbots in customer service situations is the fact that one chatbot can handle many customer cases simultaneously, whereas a human operator can only handle one at a time.

3.1.3 Benefits of Conversational Agents

CAs, in particular chatbots and voicebots, have proved to provide a lot of benefits to the modern world. One of the main benefits of CAs is their 24/7 availability. Unlike human customer service representatives which are available only a limited number of hours during the day, CAs provide fast responses to customers all round the clock. This has a great advantage for companies to increase their online sales. Another advantage is that CAs can reduce customer's searching time. The amount of time users spend navigating various websites in search for products/services can be greatly reduced if users could be guided to their search by CIs. This can contribute to an increased customer's satisfaction which is of course important these days, as various businesses are in increased competition. Additionally, CAs can benefit businesses by saving them valuable time that is usually spent on repetitive tasks such as simple customer support. By using CAs to automate and provide fast simple responses, businesses can optimize better their resources. As an example, only complex queries, which cannot be resolved by the Cis can be directed to a human representative.

3.2 Early Conversational Agents

The Turing Test (TT): Back in 1950, Turing developed the idea of what is known today as the TT. It is based on a game called the imitation game, where initially a player (the interrogator) had to investigate who of two people was the woman and who was the man. The goal of the game was for both, man and woman, to convince the interrogator of being the woman. The interaction between interrogator and the other players was performed in a written form. Turing in 1950 proposed to replace one player in this game with a computer and ask the interrogator to decide which of the conversation partners was human and which of them was not. This idea essentially described computer programs that communicate with humans using natural language, and interact in such a way, that it can be difficult to distinguish the human from the machine. At the time Turing described the TT, no actual software system existed that would be used in that experiment. It was sixteen years later when the first actual computer program was developed that was able to interact with in natural language.

ELIZA: In 1966, Weizenbaum published an article called “ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine”, where he proposed a program that was able to communicate with a human. ELIZA was created to be configurable by a set of scripts. Depending on the specific input from user, the computer would decide its response, leading thus to a dynamic conversation path “controlled” by the words of the human. ELIZA was able to achieve that behaviour by analysing the input and select a ruling topic. As an example, for the sentence “Yesterday, I had dinner with my mother”, the term mother could have been extracted and used to provide a response like "Tell me more about your mother". By selecting a word from the input, ELIZA gave to the user the impression of being understood.

PARRY: Colby et al. developed another important CA named PARRY. This CA was designed to simulate the behaviour of a paranoid patient. To validate the quality of their program the authors implemented a variation of the TT. Two versions of the CA were developed. The first with stronger and one with weaker paranoid tendencies. Using these models the authors conducted two different tests. The first test was conducted by letting judges interview versions of each (strong/weak) model and an actual patient diagnosed with paranoia. In a follow-up test, psychiatrists were provided with interview protocols from a patient with paranoia and the computer model. The psychiatrist had to decide which protocol belongs to the human and which to the computer program, respectively. In this test, 52% of the decisions were right. As a conclusion, it was assumed that psychiatrists were not able to distinguish between the human patient and the computer program as long as only written communication was used.

The Loebner Prize: The Loebner Prize was introduced as an annual competition to discover the most human-like chatbot by implementing a TT. Since 1991, a bronze medal is awarded to the CA that outperforms the other participants. A silver medal and USD 25,000 is awarded to the first system that can convince the majority of the judges to be human. A winner of the gold medal and USD 100,000 needs to additionally handle audio and video input (Powers 2002: 3).

AIML: A.L.I.C.E: The Artificial Intelligence Markup Language (AIML) was created by Wallace in collaboration with various volunteers. AIML was designed as a derivative of the eXtensible Markup Language (XML) and is used to store the knowledge of a CA. In simple terms, AIML allows a "botmaster" to specify a set of rules for how to react on a specific input. In AIML language, a pattern may contain either of words, spaces and two wildcard symbols, _ and *. The pattern is case insensitive. The first CA that used AIML was the Artificial Linguistic Internet Computer Entity (A.L.I.C.E) system, a Loebner prize winner for 2000, 2001 and 2004.

ChatScript: Suzette: Wilcox disapproved some mistakes in AIML. Initially, he considered that AIML may requires too many "redundant" rules, thus a not-optimised system. The other issue was related to the specific "input" pattern of AIML. According to his opinion, the pattern matching could have included the use of reusable synonym definitions. Based on these conclusions, Wilcox designed an alternative to AIML system, by extending the CHAT-L engine and created the Chatscript. The first ChatScript-based CA that won the Loebner Prize Competition was Suzette in 2010.

Data-driven Approaches: Although all these previous attempts, like ELIZA, PARRY, A.L.I.C.E, were rule-based systems, there are other alternative CA approaches that utilise ML algorithms to extract information from data in order to understand and reply to user queries. A system that was able to understand and reply to twitter messages was proposed by Ritter, Cherry, and Dolan. A technology called statistical machine translation (SMT) was used for the development of their proposed system. SMT works by modelling the probability of a response to an input. All these probabilities can be learned from a training dataset. This is a different from rule-based approach, as knowledge is extracted from existing data.

3.2.1 Evaluation of existing chatbots

Currently, there is a lack of a common and widely acceptable metric framework for chatbot evaluation. One of the most extensively used chatbot evaluation frameworks is PARAdigm for DIalogue System Evaluation (PARADISE). The model distinguishes the needs the chatbot requires in order to fulfil a task from the actual produced bot's outcome.

The framework via questionnaires attempts to collect users ratings, and from this input, it evaluates subjective factors such as ease of usage, naturalness, friendliness, or even willingness to use the system again. PARADISE also manages to evaluate the efficiency of the chatbot via maximizing task success while in the same time minimizing dialogue costs. These dialogue costs are defined either as efficiency costs (total elapsed time, number of systems turns, total number of system turns per task, and total elapsed time per turn) or qualitative costs (number of re-prompts, number of inappropriate system responses, concept accuracy, turn correction ratio). Hung, Elvir, Gonzalez, DeMara (2009) used PARADISE to evaluate the effectiveness of a chatbot, by checking how well it can sustain the natural flow of the conversation. In order to achieve such evaluation, authors assumed efficiency costs as resource consumption used to complete a particular task and quality costs as the content of the conversation. Kuligowska (2015) alternatively proposed a number of metrics for chatbot evaluation as they were applied in the business sector. The proposed evaluation was applied to 29 polish- speaking chatbots using a 5-point rating scale. The following features were considered in this evaluation: visual appearance, implementation style of the website, speech abilities, knowledge-base, knowledge presentation and its usability (click-through links, ability to scroll through past dialogues), conversational capabilities, personality, personalization options, responses in unforeseen situations and possibility for users to rate the chatbot and website.

3.3 Designing a chatbot with cognitive characteristics

Currently, one well known framework used for the development of autonomous agents with human-like behaviour is the BDI model. This model has been constructed based on three concepts: beliefs, desire, and intention. Agents can draw conclusions based on “beliefs” which represent the individual’s knowledge about the environment as well as their own internal state. “Desires” represent the specific goals which the individual has set as a target, while “intentions” represent the set of plans or sequence of actions which the individual intends to follow in order to meet the target. The BDI model has been successfully evaluated in creating human-like characters. However, although the model is an abstraction of human deliberation, it is lacking many generic aspects of human behaviour and reasoning (Wong, Thangarajah, Padgham 2012: 2-3).

When developers design a human-like chatbot, it may be critical for them to consider which features could influence the UX to full the user’s needs. More specifically, UX can be may be defined by the following three components: (i) its usefulness, which “measures” how well an artefact enables a user to reach a specific target; (ii) its usability, which “checks” how easy to use an artefact; and (iii) satisfaction, which “measures” how well satisfied a user is with such interaction with an artefact. One feature of UX in the framework of AI is to show how human-like chatbots should behave, and how this aspect affects UX. Although existing research studies have investigated how chatbots could be evolved to converse or behave in a human-like manner, there is clear answer on whether this is the proper methodology to move forward from a UX perspective of interacting with chatbots towards such behaviour. There is a risk that a possible emotional reaction from a chatbot may not match with the conventions and “rules” of a human conversation, and this may create some type of discomfort to users. An additional possible risk in such cases, is that such emotional intelligence may mislead users that these systems are more capable than they actually are, leading them to false expectations. The users could also understand the chatbots as being more sensible in producing their responses. Emotional responses that are related to a situation that has been shown to create a reliable impression, enable users to trust chatbots as systems capable of performing such responses. It has been claimed that a chatbot can become more trustworthy if it is more like robotic than human-like (Duijst 2017: 4).

However, a machine does not need to be human-like to be appealing, and obviously it is not necessary to have chatbots that are human-like and emotionally intelligent simultaneously. In some cases, users created a rather negative response when they were with interaction with human-like chatbots. However, studies revealed that in general, users are fascinated to human-like chatbots that includes aspects of AI that allows for a human-like use of language.

The main design issue is to "create" a trust in chatbots, which is depends on the design of personality as well as the design of the conversation. Features that can contribute to users trusting chatbots are: (i) small talk; (ii) a chatbot being open and transparent of its role; (iii) the ability to react and respond to signals that could reveal emotional and social behaviour; and finally (iv) an authentic chatbot that matches the user on a personal level and in conversation. In this case, authentic means that the chatbot can act as a person with values, attitude and culture. Finally, the following features need to be considered when we design a chatbot. (i) To clarify the chatbot's competence and abilities; (ii) to have the ability to maintain the context through a conversation; (iii) to be able to manage adequately a failed dialogue; and (iv) finally to terminate a conversation in an appropriate way. It has been shown that users prefer chatbots that communicate in a human-like manner (e.g., have an understanding of negative statements, able to manage failure and able to ask intelligent questions) and such trend has been adopted in current systems as seen in messaging conversations (Svenningsson, Faraon 2019: 4).

Chapter 4
Chatbot architecture

The aim of this chapter is to introduce some basic methods and techniques behind a CA. These methods are applied by most agents nowadays and follow the same processing pattern. First of all, a brief introduction to data pre-processing will be presented in order to facilitate the way in which textual features are extracted from a question. Data pre-processing is an important part and is the first step in the process of text-analysis. Depending on the structure of the data and the type of analysis, certain techniques been developed with the aim of extracting knowledge from data. The most important of these methods will be presented below.

4.1 Data Science in Chatbots

Classification is considered as one of the basic data mining techniques. This method is based on examining the properties of an object and assigning them based on a predefined set of classes. The objects/samples that will be categorized are database entries and this process has to allocate these entries to some predefined categories. In most cases, there is a limited number of categories and each entry has to be successfully assigned to an appropriate category.

Clustering is the method of separating a dataset into a set of clusters. The difference between clustering and categorization is that the former is not based on predefined categories. Data are organized in clusters based on the similarity between them, utilising some similarity criteria

Association Rules is one of important data mining processes. What makes this process particularly interesting is the concise way in which useful information presented, which is easily understood by users. Association rules reveal hidden "correlations" between the features of a data set. These correlations are presented in the following form: A → B, where A and B refer to the feature sets present in the data under analysis.

Regression is a topic that has been studied extensively in statistics and neural networks. The main purpose here is to to predict the outcome of a dependent variable, based on a number of independent attributes/features which are related to the specific case study. We usually use a model for the variable. Regression covers a large part of the data mining sector that deals with forecasting.

Decision trees have been extensively studied and are widely used in both classification and regression. A decision tree represents a set of IF-THEN rules starting at the root of the tree and ending at its leaves. The internal nodes of a decision tree represent the features of the problem, the edges the possible values of these features, and leafs represent the possible classes of the problem. More specifically, let's suppose we have a set of records and a list of attributes for each of them. A decision tree, in all records, is a tree where each node corresponds to the name of a Xi attribute, each edge is named with a predicate that can be applied to the attribute that constitutes the node name and each leaf denotes a class. Starting from the root of the tree, the algorithm of the technique creates subsets based on the optimal feature. For each of these subsets, the above procedure is repeated constantly using the remaining features as criteria for creating new subsets. This process stops when all the instances of the subset belong to the same class.

Neural networks are a technique that can be applied to prediction, classification and segmentation. Their ability to learn from data, mimics humans' ability to learn from their experiences. A neural network that is designed for a problem can make valid predictions for new instances of that problem. Neural networks use a set of processing elements (nodes), analogous to the neurons in the human mind. These nodes are interconnected in a network that can recognize patterns as soon as they are presented within a data set. That is, the network can learn from experience just like humans do. Usually, neural networks work in such a way that a particular input leads to a specific output. Neural networks are very powerful tools, with highly satisfactory performance even in non-classical cases of Data Mining problems. They also have a very high tolerance for incomplete data or "noise" data. For this reason, they are widely used despite the fact that they are difficult to interpret.

4.2 NLP Techniques

Tokenization: In this process, larger pieces of text are broken down into smaller pieces (tokens). For example, large sections of text are divided into sentences, sentences into words, and so on. In fact, the application of this technique is not always so simple, as there are often mistakes, so it's not always clear when the end of a word or sentence is marked. But even when there are no spelling mistakes, punctuation marks often create ambiguity as to how it should be segmented.

Stemming: It's the process of reducing a word to its simplest form, ie the technique by which the ending of a word is removed in order to take its root (stem) and is applied in order to reduce the size of the vocabulary that is about to be processed. However, sometimes the cut of the word does not correspond conceptually to its original meaning, while in others, the returned word is not real and creates confusion. There are three algorithms for stemming: Truncating methods, statistical methods and mixed methods. Truncating methods are used to remove the prefix or suffix of a word. Statistical methods remove suffixes, having previously applied a statistical procedure. Mixed methods include morphological analysis of syntactic variations on different gradients, plurals, and analysis of derivatives associated with part of speech in a sentence. One of the most popular algorithms is Porter Stemmer, which was proposed in 1980. It belongs to the category of truncating methods and is based on the idea that English suffixes consist of other smaller and simpler ones.

Lemmatization: The logic and basic function of this technique is generally the same as that of stemming. Similarly, lemmatization reduces the different forms of a word and restores them to their lemma. The difference between these two methods lies in the fact that in the case of lemmatization, the syntactic function and the meaning of the word are taken into consideration, something that stemming ignores. Therefore, using lemmatization, the algorithm manages to return as a result the linguistically correct root of each word and this result is an existing word.

Stopwords removal: In general, data pre-processing aims to modify the text in such a way that feature extraction can be applied. But beyond that, it aims to eliminate useless information. Stopwords are part of natural language and are words that do not offer anything in the meaning of the text, such as articles, intentions, pronouns, etc. Removing these words also reduces the complexity of the problem. There are four methods for removing stopwords. According to the classic method, stopwords are removed, based on a pre-defined list. In Z-method, in addition to the previous list, we use three methods: Remove the most frequently words in general, remove the words that appear only once, and remove the words that appear frequently throughout the text. In Mutual Information method, the common information between a word and a class is calculated and if it is low, it means that the word does not provide specific information for the class and it is suggested to remove it. In the Term Based Random Sampling method, the terms are classified based on the Kullback-Leibler deviation measure into randomly selected online documents, and then the list of stopwords is constructed.

Part-Of-Speech Tagging: By this term, we mean parts of speech such as nouns, verbs, adjectives, adverbs, etc. Part-of-speech tagging is the process of noting the words of a text and assigning them to a specific part of speech based on their definition and meaning in the text. The process is done with the synergy of algorithms that fall into two categories, rule-based and probability algorithms. In the first case, the algorithm receives as input a sequence of segmented words and a tagset and returns the word sequence, where each word is accompanied by an appropriate tag. In the second case, the algorithm is based on models such as Hidden Markov Model (HMM). Tag rendering is a clarifying function, as a word can be associated with more than one tag. So the aim is to choose the right tag depending on the meaning of the text. The method of part-of-speech tagging serves the best investigation of a sentence from a linguistic point of view. One of the most important collections of English tags is the Penn Treebank tagset, which has been used to tag many text corpora.

Name Entity Recognition (NER): The first step in Information Extraction (IE) is to locate the entities within a text. A name entity can be anything that might be referred to by an appropriate name, such as a person, a location or an organization. However, this term has been extended to include objects that are not generally name entities, such as dates or other time expressions, and even numeric expressions such as values. In addition to extracting events and the relationships between them, nominal entities are useful for other language processing tasks. For example, in sentiment analysis, there is a need to know a consumer's feeling towards a particular entity. NER is the finding of spaces within text that make up regular names and then the classification of the entity type. Entity recognition is a difficult process because of the ambiguity of segmentation; it must be decided what is considered an entity and what is not, as well as what are the boundaries. Another difficulty is encountered by ambiguity. The standard algorithm for naming entities is the word-for-word sequential process, in which the specified names include both the formula and the boundaries. A sequential classifier, such as a MEMM (Maximum Entropy Markov Model) or CRF (Conditional Random Field), is trained to name tokens in a text, with labels indicating the presence of specific types of named entities.

N-Grams: N-grams in Computational Linguistics and Probability is a sequence of n objects from a given text or voice sequence. These objects can be phonems, syllables, letters, words. The n-grams of size 1 are known as unigrams, the n-grams of size 2 are known as bigrams, and the n-grams of size 3 are called trigrams. N-gram models are linguistic models whose purpose is to predict the next object in a sequence of order (n - 1) of Markov models. There are different Markov models, depending on the case we use. In this case, we are talking about Markov Chains named after Andrey Markov, according to which, a Markov chain is a random process, which undergoes transitions from one state to another. These chains should be characterized as memoryless. This practically means that the distribution of probabilities in the next situation depends only on the current situation and not on the sequence of events that preceded it. The two main advantages of n-gram models are their relative simplicity and the ability to upgrade by simply increasing n. These models can be used to store more contexts at the cost of later execution, but with less memory, allowing small experiments to be effective.

Bag of Words: Bag of Words (BoW) is an algorithm that measures how many times a word appears in a document. These numbers allow us to compare documents and measure their similarities for applications such as search, document classification and topic modeling. In this approach, documents are represented as "word bags". Combined with various machine learning techniques, this type of technique has repeatedly been shown to be effective and robust in classifying text documents (e.g., to determine if a piece of text has a positive or negative emotion).

4.3 Linguistic Techniques

Conversational agents are based on sentences, which means that the input, given by the user, usually consists of sentences. These sentences must be analyzed by an agent, so that after a specific process, they can be interpreted correctly and a relevant answer can be given. For this reason, it is considered appropriate to take a linguistic approach to sentence processing and review some steps that are used by agents to analyze a user's input. In order to implement a program that understands and analyzes natural language, one of the basic things to keep in mind is to define precisely the ultimate goal as well as the final form of internal representation. Before examining in detail the natural language techniques from a computer science approach it is useful to make a reference in the linguistic approach of sentence processing.

Morphological Analysis: Each word is analyzed in its components as well as non-verbal symbols, such as punctuation marks that are separated from words. The morphological analysis should make a distinguish between, for example, the word "Jim's", the noun "Jim" and the possessive suffix "'s". Additionally, the process will assign each word of the sentence to syntactic categories. This is usually done at this point, because the interpretation of suffixes or prefixes may be based on the syntactic category of each word. For example, consider the word "prints". This is either a noun (with "-s" denoting the plural) or a verb in the third person singular (as in "he prints").

Syntactic Analysis: Linear word sequences are converted into structures that illustrate how words relate to each other. Some word sequences may be rejected if they violate the rules of linguistics, which regulate the way words are combined. Syntactic analysis should take the results of the morphological analysis to make a structured description of the sentence. The purpose of this parsing process is to convert a linear list of words that make up a sentence into a structure that defines each section represented by that list. A linear sentence is transformed into a hierarchical structure, and this structure corresponds to sentence units, such as noun phrases. These sections will correspond to semantic sections when performing the semantic analysis.

Semantic Analysis: This process gives meaning to the structures that are extracted by the syntactic analyzer. In other words, it becomes a kind of illustration between the syntactic structures and the objects of the problem area. Structures that cannot be assigned will be discarded. Semantic analysis must depict individual words in the appropriate objects of the database or knowledge base and create the correct structures that correspond to the way in which the meanings of the individual words are combined with each other.

Discourse Integration: The meaning of a particular sentence may be based on the preceding sentences and may affect the meaning of the following sentences. At this point after the above steps, we have ascertained what this sentence is talking about, but we don't know yet which specific subjects it refers to. Through this process it is possible to cross for example that "I" refers to a specific person.

Pragmatic Analysis: The structure that represents what has been said is reinterpreted to determine what it really means. So far we have a clear description of what has been said, according to what has been provided by our knowledge base. The final step in achieving an effective understanding process is to decide what to do about the outcome. One possible way is to record what has been said and accept it as an existing fact. For those proposals, the intended result of what is clearly stated, is exactly what is needed. But for other proposals, the desired result is different. We can reveal this by applying rules that characterize a dialogue. The final step in pragmatic analysis is the transformation, whenever necessary, from the representation of the knowledge base to a command to be executed by the system.

Syntactic processing (Parsing): Syntactic processing is the step in which an input sentence is transformed into a hierarchical structure that responds to the semantic units of the sentence. The whole process is also called parsing. Although there are natural language comprehension systems that omit it, analysis plays an important role for two reasons: Semantic processing must handle the components of the sentence. If there is no syntactic analysis, then the system dealing with semantics must decide on its own components. When the analysis is finished, it limits the number of components that semantics has already considered. Syntactic analysis requires less calculations than semantic analysis (which requires considerable effort to draw a conclusion). For this reason it has an important role in reducing the overall complexity of the system.

Although there are many ways to generate a parse, almost all systems used have two main components: An explicit representation, called grammar, of the syntactic rules of language and a process, called parser, that compares grammar with input sentences to produce syntactic structures.

Grammars: The most common way to represent grammar is a set of production rules. Although the details of the permissible forms of the rules differ, the basic idea is a context-free grammar. The first rule suggest that "A sentence consists of a noun phrase, is followed by a verb phrase". In this grammar, the vertical bar represents the logical "or". $\in$ indicates the empty string.

Grammar formalities are subject to many linguistic theories. It should be noted that there is a general agreement that pure context-free grammars are not effective in describing natural languages. Regardless of the theoretical basis of grammar, the process of analysis takes the rules of grammar and compares them with the input sentence. Each matching rule adds something to complete the structure of the sentence. The simplest structure that can be created is a parse tree, which records the rules and how they fit together. Each node of the syntax tree corresponds to either an input word or a non-terminal grammar symbol. Each level of a tree corresponds to a specific grammar rule. Therefore, we observe that a grammar defines two things for a language:

- Its low production capacity, by which we mean a set of sentences contained within the language. This set of grammatical sentences consists of exactly those sentences that can match a set of grammar rules.
- Its strong productive capacity, by which we mean the structure attached to any grammatically correct sentence of the language.

4.3.1 Top-Down and Bottom-Up Anaysis

To analyze a sentence, we need to find a way to form the sentence. There are two ways to do this:

- Top Down Analysis (From Top to Bottom) - Starting with an initial symbol, we apply the grammar rules forward, until the symbols at the end points of the tree are matched with appropriate parts of the sentence.

- Bottom Up Analysis (From Bottom to Top) - Starting from the sentence under analysis, we apply the grammar rules in reverse, until a unique tree is created, whose endpoints are words of the sentence and its root is the initial symbol.

Chapter 5
Related Work

In this chapter, we will present some of the conversational agents that are currently available online and how they work individually. Most agents, as will be seen below, have identical characteristics and act in the same way. Some key terms that are used in the development and operation of conversational agents are presented below:

Utterance is anything the user asks the conversational agent. For example, when a user says "I want to buy a smartphone", this whole sentence is the utterance. Intent is when a user interacts with a conversational agent, expresses a purpose and the system recognizes what the user wants to do. The user's message is parsed into the NLU engine and the intent is extracted to match an action. For example, when a user says "I want to buy a smartphone" the intention could be "buy_smartphone". Entities represent concepts related to the object served by the conversational agent and are used to export parameters and metadata through a natural language text. In the training of a conversational agent, multiple intentions can be directly related to one entity and many entities can exist in one expression. For example in the message "I want to book a ticket for a movie" the entity is a movie, otherwise "movie". Entities can represent a species or a quantity. In Action, once the user's intent has been detected and associated with an entity, the agent must respond appropriately to the user by performing an action through programming methods. For example, when a user asks "What will the weather be like tomorrow?" the conversational agent should connect to an API or database, retrieve the data, edit it, and then present it to the user in a comprehensible response format (Tamrakar, Wani 2021: 8).

5.1 Wit.AI

Wit.ai is a platform offered for free by Facebook. Its creation and editing can be performed from the website of the platform. Using natural language editing tools it can process the information provided by each user but also to extract decisions/conclusions from it for further use. The primary interface that the user sees when the user interacts with the platform and logs in is the following:

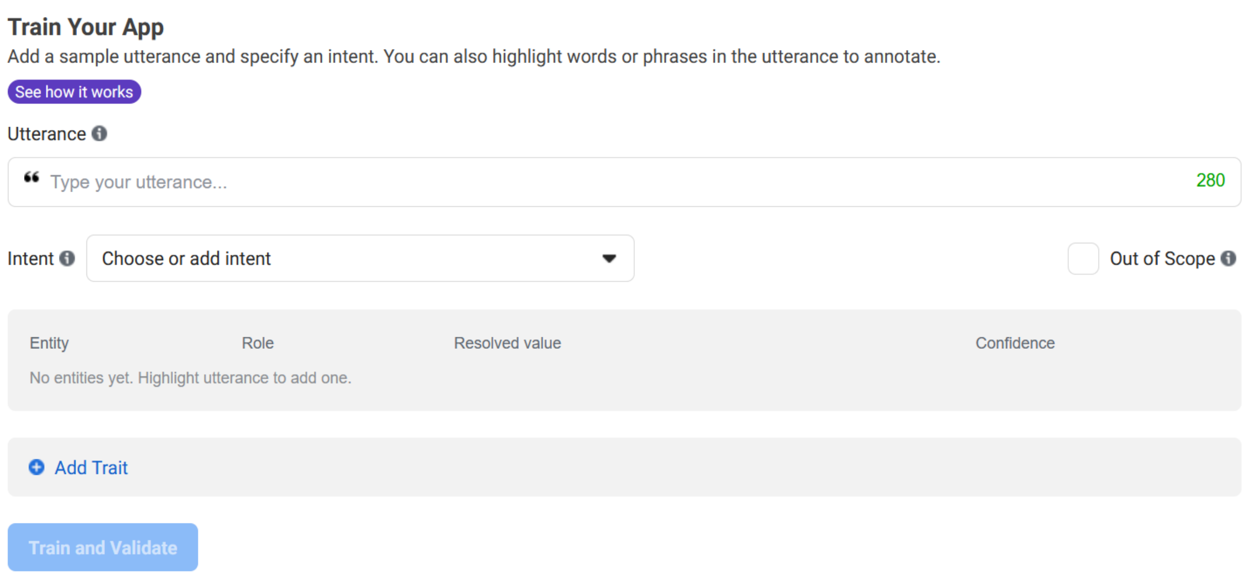


Figure 3: Wit.AI settings

This bot must be trained to work, and the administrator of each bot is responsible for this. Although the platform itself provides the ability to identify a large number of entities, the administrator has the capability to train the bot to identify entities from specific circumstances and categories. To train the bot, the administrator enters keyword phrases in the utterance field and then associates keywords in the intent field. If one of the default intentions of the system is not fully reflected, the user can add an Intent. An example is given below and its results are presented:

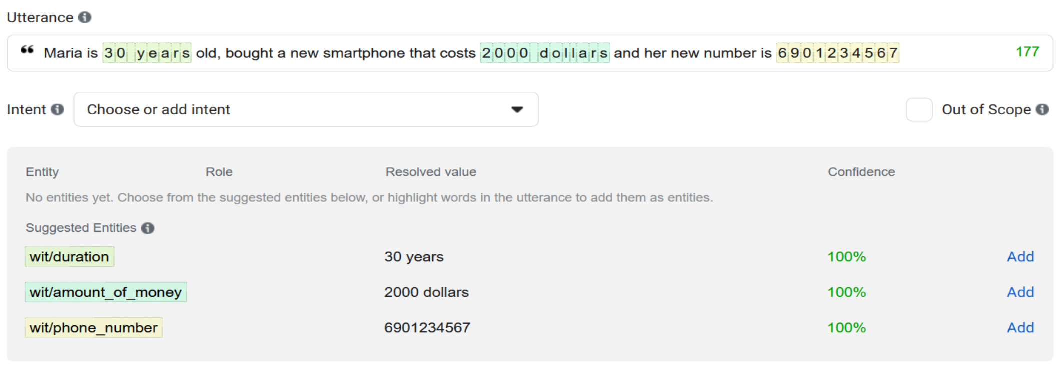


Figure 4: Wit.AI settings

In the above example, the sentence "Maria is 30 years old, bought a new smartphone that costs 2000 dollars and her new number is 6901234567" was provided as input to WitAI, and the system was able to associate the "Duration" with the phrase "30 years", the "amount of money" with the phrase "2000 dollars" and finally the "6901234567" with "phone number". This recognition was performed without the intervention of the bot administrator, using the information/data the bot had acquired during its setup process. However, the user is able to select words or phrases of the sentences and add new intents, enriching thus system's database. For example, we could define the phrase "smartphone" as an "electronic device". Obviously, the recognition accuracy of the system for future tasks can be improved by providing to the system's database adequate quantity of relevant information/data. In practice, this specific bot can be utilized for user's interaction, by incorporating it into with Facebook Messenger or any other similar application the developer has created (Biswas 2018: 67-70).

5.2 Chatfuel

This program offers the necessary “bot”-related capabilities through the Facebook Messenger platform. This “tight” interaction/connection of the chatbot to the specific social network, although reveals several limitations compared to the existing competition, which often operates on many different platforms, at the same time it has some advantages.

The integration of the bot with the Facebook Messenger allows the creator to access some information of the user that could not have been available under different circumstances, such as his/her name and/or age. To take advantage of this information, Chatfuel thus retrieve user’s profile and store into a variable and during the operation, the bot can address the user with his/her name, as an example. Another important feature of Chatfuel is the rather limited usage of the so-called "artificial intelligence" (AI). As this tool operates via Messenger, the user has the ability to interact with it, not only through the preset keys displayed during dialogue frame, but also by typing in the context of the conversation.

The creator thus has the ability to define specific phrases and words which could be linked to a corresponding action inside the chatbot so that it can execute, if the user enter one of them. For example if during a conversation, a user enter the word "Contact", the program will be able to skip this dialog box and refer it directly to corresponding tab with the contact details. The particular ability of creating an AI chatbot, is present only on the Chatfuel platform and can make the bot quite "smart" as with the corresponding investment of time. Also, integration of "Keywords" are able to make it recognize a huge range expressions and react accordingly.

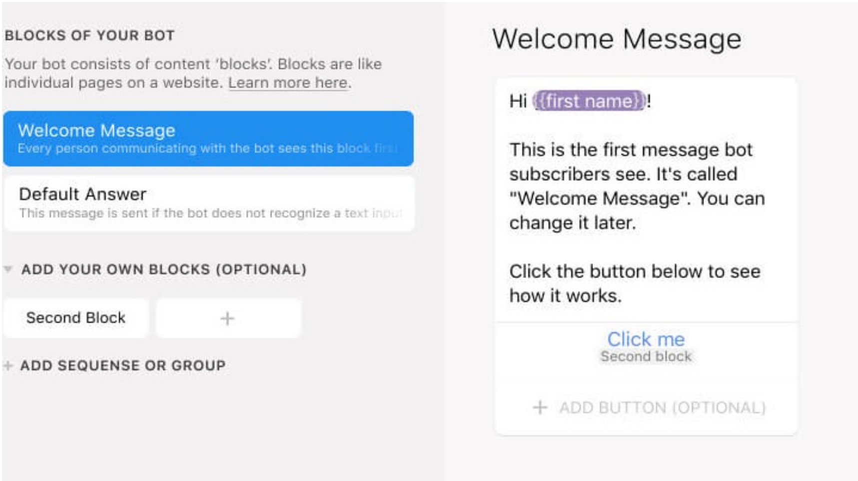


Figure 5: Chatfuel interface

Regarding some restrictions encountered when using Chatfuel, perhaps the most important of them deals with the use of keys to navigate the user during a conversation. It is clear that the ability to type questions provides a significant degree of freedom during the dialogue. However, most users are not expected to know exactly what will follow when asking their question to the bot. This automatically means that sometimes, asking questions directly is not very efficient. This issue is solved by the use of navigation-keys, which serve as a "guide" in the conversation and provide instructions to direct the users to the corresponding part of the dialogue that concerns them. Although Chatfuel provides the ability to create such keys, unlike the other platforms examined, the maximum number of choices that can occur simultaneously is three. This practically means that it is impossible to present more than three options to the user at the same time, which significantly limits the process of creating a digital assistant that aims to collect information from the user.

5.3 Pandorasbots

Pandorabots is one of the oldest but also one of the most advanced conversational agent platforms. It specializes in building chatbots using the "AIML" language and is an open source software. What distinguishes it from other platforms is that the administrator creates the chatbot by writing programming code rather than using a graphical interface. Due to this fact, it is much more difficult to build a chatbot through it, since its creator must have basic programming knowledge but also know in depth "AIML".

Pandorabots is an AI company that manages chatbot development services. This platform has some of its resources free of charge with licenses such as the Global Public License (GPL) or through APIs. It incorporates several machine learning technologies each with their own limitations and peculiarities. Pandorabots can reside locally on a device and continue to work offline, and it is able to learn and retain user details. It is quite easy to develop a pandorabot, as it is possible to easily integrate data content from external APIs and external databases. An important point is that Pandorabots is multilingual and is able to work in any environment. Pandorabots currently uses Google's API to recognize speech, but other speech models can be easily integrated.

5.4 Quriobot

One of the great advantages of Quriobot is that the conversational agent can be shared in many different ways. Specifically, the administrator of this agent has the ability to integrate it with applications such as Whatsapp and Facebook Messenger, while it can be placed directly on a website, as it is not needed to use a third party application. This makes this platform extremely flexible. Another feature worth mentioning about Quriobot is the existence of of free templates that can be used as a basis, that a developer can be based on to implement his/her own chatbot. Finally, Quriobot stands out from its competitors, as it is the only platform which offers the possibility of visualization. In particular, the creator has the ability to work on the program, and display it as a flowchart. In other words, he/she has the ability to perform "visual programming", connecting the steps with each other and defining their relationships through variables.

5.5 IBM Watson

Another well-known conversational agent is IBM Watson. IBM entered in this area in 2006 as more and more companies invested in NLP industry (Biswas 2018: 101-104). Like any conversational agent, Watson takes natural language as input and is able to answer questions. In fact, what set Watson apart from the rest of conversational agents of its time, is that it can perform the exact opposite process. More specifically, in 2011, in the tv-show Jeopardy, the presenter gave some data to the players, and they in turn tried to find out what is the question that corresponds to a specific answer. IBM Watson has received rave reviews from its presence.

In particular, IBM Watson is a Question - Answering system. In other words, it has the ability to answer questions that are asked in natural language. Watson Assistant later has evolved into a system that is used by businesses to provide personalized customer experience and improve existing human-computer interaction. Developers and computer technicians from companies can integrate IBM Watson Assistant to their business in order to create from simple chatbots to complex business solutions for customer service. IBM Watson Assistant can be applied in a wide variety of business, tourism, health and education. Indicatively, during the pandemic, IBM Watson Assistant for Citizen is in great demand and is programmed to answer questions and help the general public with issues related to Covid-19. In addition, the IBM Watson Assistant for Automotive was designed to enhance the car experience and its interaction with passengers.

From a technical point of view, IBM Watson uses large-scale algorithms based on artificial intelligence to extract maximum information from minimal inputs. It is able to accept as input the question of a user, and as a supercomputer, to utilize a wide range of knowledge given to it, to give the appropriate answer. Its mode of operation is similar to that of most conversational agents. More specifically, it accepts as input the data from the users and the system looks back at the data it already has. It then separates the words that have been formulated and tries to identify keywords that are related to the knowledge of the system, in order to identify common elements. In the next stage, assumptions are made and finally gives an answer based on an assessment it has made.

IBM Watson is not a static program, but it is evolving as its administrator can constantly train it with new information so that the given answers are more accurate. A user who wants to create an IBM agent can do so from its website.

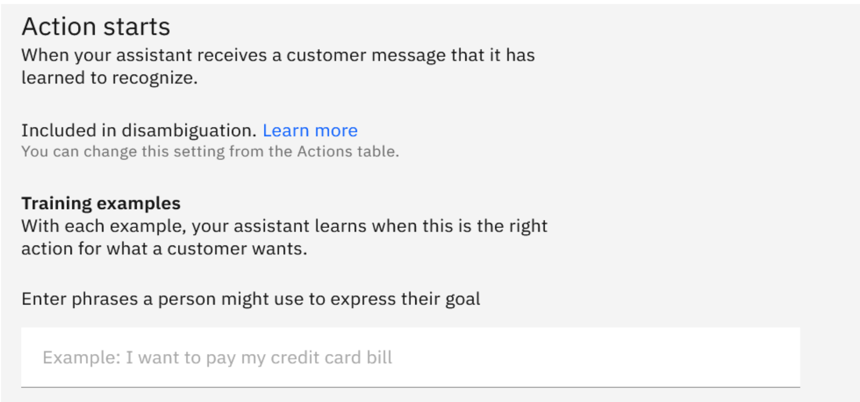


Figure 6: IBM Watson settings

In the image above, the user can train the conversational agent by filling predifined questions and specific answers that will match the questions. Finally, IBM Watson allow us to define certain conditions of the IF → ELSE format.

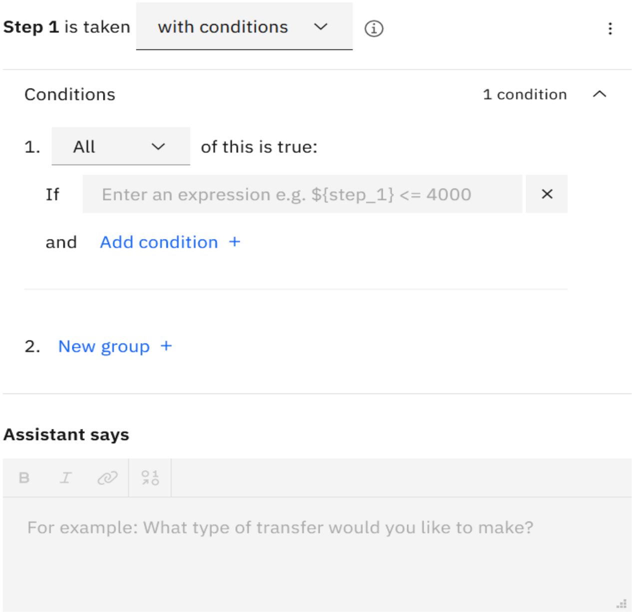


Figure 7: IBM Watson – Defining Conditions

5.6 Snatchbot

Snatcbot is a Platform-as-a-service, which allows users to design, test and develop high quality bots without having to develop their own infrastructure. SnatchBot can be used with almost any communication channel, such as text, voice and email or social network applications such as Facebook, Skype, Slack, SnatchApp, corporate websites and mobile applications. It has flexibility and it's easy to use it, even if one has limited or no programming knowledge**.** It can easily be connected to user backend systems via the snactbot API, Json API or third party APIs such as Integromat or Zappier. It incorporates NLP technologies through intents and entities as well as ML for its training. It also uses Supervised ML in order to examine whether the bot needs additional training. It allows instant monitoring of chatbots usage statistics as well as activity logs. Moreover, Snatchbot offers ready-made templates for customization by users in order to integrate them into software applications. Snatcbot answers are specific and clear, as compared to many other Chatbot platforms, and users seem to be satisfied with the given answers by the agent.

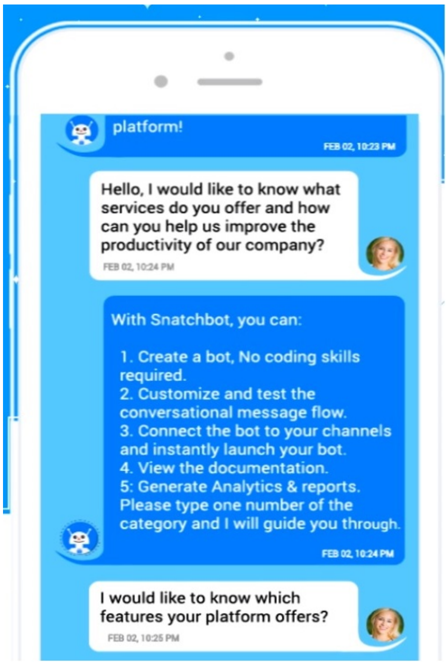


Figure 8: Snatchbot Interface

5.7 Dialogflow

Dialogflow is a natural language API that enables technological interaction between humans and computers through dialogs. Its main function is the processing and recognition of natural language, to extract the meaning of a sentence (intent). Dialogflow allows developers to include voice and written natural language interfaces in their applications (Biswas 2018: 85).

Conversational agents that are built through this platform are able to support a very high level of discussion, both written and orally. This is due to the high level of language processing and machine learning models provided by the platform. In other words, it's a platform that can respond perfectly to applications with more requirements. However, designing a chatbot through Dialogflow requires an administrator who has basic programming knowledge, as it is not so easy comparing it with other platforms.

Dialogflow is the platform provided by Google and through it, developers and businesses can easily create chatbots. In this platform, the user's question is converted into data that corresponds to specific actions and returns a response in JSON format.

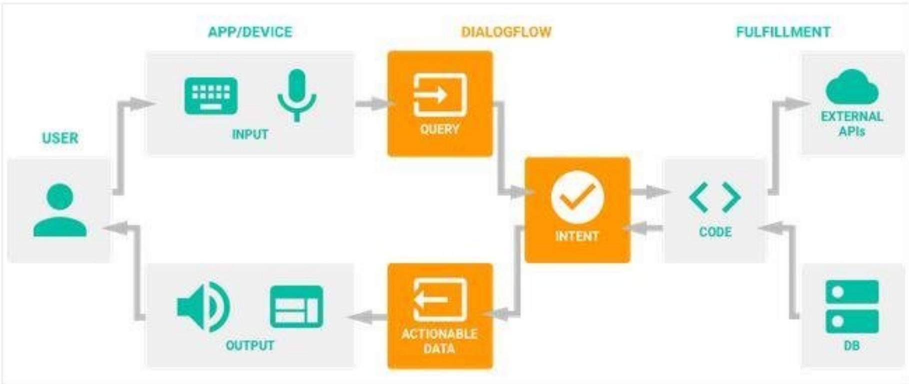


Figure 9: DialogFlow stucture

5.8 Rasa

One of the most well-known conversational agents is Rasa. It is an open source tool based on artificial intelligence technologies. Although it is free to use by the general public, it also offers the possibility of purchasing an Enterprise package that offers more benefits to the agent administrator, such as statistics on the number of messages, the duration of the conversation and the number of users who interact with the agent.

The components of Rasa are Rasa NLU (Natural Language Understanding) and Rasa Core. When a message is received, it goes first through the Rasa NLU and then to Rasa Core for further processing.

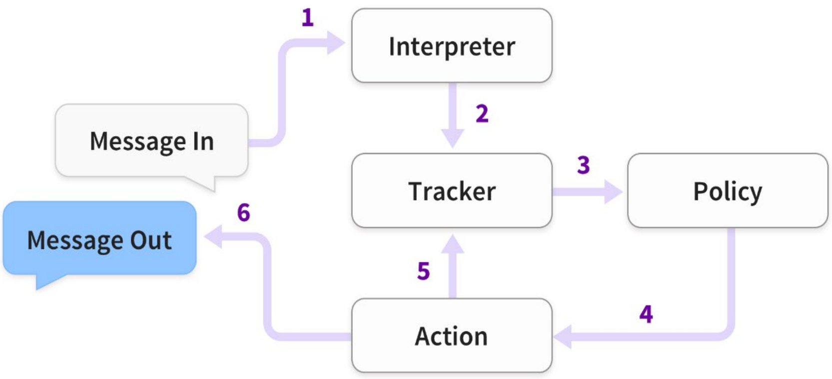


Figure 10: Rasa NLU

More specifically, Rasa NLU is responsible for understanding a user's input, ie the questions he/she is going to ask. According to the image above, Rasa NLU is only responsible for step "1", where the intent, entities, and any other structured information are extracted. The "Interpreter" is the main interface between user input and the system. The adequate understanding of user's data is the most basic step for the operation of this conversational agent, as it determines the “path” of the discussion that it is going to be developed. Whenever an agent receives data from a user, the agent should refer to the information it already has (or has been trained) to direct the discussion in a logical sequence. The above procedure is referred to as "intent classification", i.e. what is the user's intention regarding the question he/she asks. Then, the next step that is performed by the Interpreter is "Entity Extraction", which means the ability of the agent to isolate keywords from the user's message, which are crucial for the response that the user must receive.

Rasa Core, on the other hand, takes steps 2 to 6. Rasa Core is responsible for the logical path of the discussion and decides what action the bot should follow. In step 2, "Tracker" is notified that a message has been received for processing and monitors the incoming information. In the 3rd step, the "Policy", the status received by the Tracker is confirmed and it is decided which step to follow along the way. The decision made by Policy remains in the Tracker as a log, and the final answer is given to the user (Sharma 2020: 1-3).

Chapter 6
Implementation

6.1 The rationale of using Rasa

In this chapter, a conversational agent will be implemented which will be able to answer simple consumer questions. Consumers, through this agent will be able to enquire for information for any product they wish to purchase or regarding the process of their order. Rasa has been chosen for the implementation of this agent. The aim of this chapter is to demonstrate how a conversational agent is implemented, by applying natural language processing techniques mentioned previously in the theoretical part of this Master's Dissertation. In the chapter, some elements of the code written to implement the agent will be also presented.

Rasa is a free open source software used to develop conversational agents. A very important feature of Rasa that offers security of users' personal data, is that the data can be optionally remained on a local server and not shared on a Cloud system, for the purpose of sharing. Another reason for choosing Rasa is related to its simplicity in terms of usage. Any user, with a little training can learn to use this software with great success. In fact, the programming knowledge needed is minimal, so this fact makes Rasa even more popular among other similar tools. Finally, Rasa, through it updates/additions and customisations, is evolved and thus offers to a growing community of users new levels of capabilities.

6.2 Installation of Python and Rasa

Python programming language was used to create this agent. Therefore, after downloading and installing Python from its official website, we install Rasa by executing the following command in the Command Prompt (in case for a windows-based computer): *python -m pip install rasa*. After Rasa's installation, we have to install the spacy library, which provides all the dependencies related to Rasa (Name Entity Recognition etc.), with the following command:

python -m pip install spacy. Then, since the agent's communication language will be English, we need to run the following commands: *python -m spacy download en_core_web_md* and *python -m spacy download en_core_web_md en*.

6.3 Rasa NLU

RASA NLU is responsible for receiving the message from the user and then via a specific analysis, the tool "tries" to understand the meaning of user's message. Usually such message is in an un-structured text form. The Database that has been created via RASA NLU, includes a list of potential messages / questions that a user may ask. After the processing of this question, the outcome is practically the attempt to structure the text and to divide it into intent and entity. Intent is related to user's intention, i.e. what exactly he/she wants to ask and what information wishes to extract from the conversational agent. For this reason, words/phrases are associated with a common users' wish and have been grouped under a common wish, the Intent.

The type or content of intent normally varies, depending on the topic/specialization of the website where the agent is located. It could be a search for a price on a product or the availability of a product in case the website is associated with a retail store. However, in many cases, the intent is not clearly defined and the agent is thus unable to recognize user's intention. The main reason for such problematic issues it related to the fact that the system's database is incomplete without much information provided during its training phase. Another important reason for agent's failure to understand the intention is the user's expression. Since the lexical richness of most languages is practically unlimited (such as in English and Greek), everyone can expresses his/her thoughts and intentions differently, often using a slang. The table below illustrates some possible questions that users would ask on a web page and their potential Intent Classification, which reveals that they have not been categorized in depth and detail.

Question by a user	**Intent Classification**

What's the price of Iphone 12?	ask_price
Is Iphone 12 available at your store?	ask_stock
How many years warranty does Iphone 12 have?	ask_warranty

After the stage of the Intent Classification, another process the "Entity Extraction" needs to be implemented. According to this processing step, RASA NLU must be able to recognize keywords that belong to a certain entity and eventually associates it with a specific item in the system's database. In the above example, the question "What's the price of Iphone 12?" the entity is the "Iphone 12" since it is a specific product that needs minimal clarification to be fully characterized/recognized (capacity and color of the device). However, from the first stage of the above process, the conversational agent understood that the user's intention is to find out the price for a specific product (ask_price_Iphone12). In a commercial agent-based system, it is obvious that for each product, a separate entity needs to be defined which will incorporate all its features, such as color and capacity of the model. However, as it shown in the above example, it is typical that the agent may not be able to fully recognize the user's intention, due to the way or style he/she enquired the specific question. More specifically, a user could have asked "What's the price of the latest Iphone?". At the time of writing of this report, the latest model is "13", so if the agent has not been properly trained, then it could be impossible to associate the "latest iphone" query with a predefined Entity in its database.

6.3.1 Training our conversational agent

In this subsection, the conversational agent will be trained with some intents. As it has been mentioned previously, the Rasa database optionally remains on a computer rather than a cloud system. Therefore, a file containing the intents that we will need to define must be created in a local folder. Rasa uses Markdown files, so the file should end in ".md". It is very important to know that this file must be placed in a folder called "data", otherwise it will not be recognized by RASA NLU for training. This file can be edited with most editors. Specifically, for this agent, Sublime Text was used, and some related screenshots are illustrated in the following lines. In order to define the user's intent, we use "## intent:" and then we specify the contents of that intent. The following code is an example of an intention for a greeting:

```
## intent: greet
- good morning
- good evening
- good afternoon
- hi
- hello
- hey
- how are you
```

Figure 11: Defining intents

In the above list, seven different ways of greetings are provided. Obviously, the list can be expanded to include more words or phrases to cover any potential user intents. Similarly, farewell intents can be defined, as shown below:

```
## intent:bye
- goodbye
- good night
- goodnight
- see you later
- talk to you later
- ok thanks bye
```

Figure 12: Defining intents

In the next step, our database must be enriched with more and more intents that are related with the topics/specialization of the conversational agent of the Master's Dissertation, that is, the customer service in an online marketplace of electronic products. The following example shows intents related to the process of an order, as contained in the created .md file. The list of questions is purely indicative of the context of this Thesis. At a professional level, a team of Conversational Designers will have to define questions of all kinds, for each product category they might have.

```
## intent: order_status
- Check status of my order
- When is my phone coming in
- I'd like to check the status of my order
- I'd like an update on my order
- I haven't received my order yet, please check it
- Where is my order?
- I want to check the status of my order
```

Figure 13: Defining intents

6.3.2 Pipeline

RASA NLU Pipeline is a necessary component for the efficient functionality of the conversational agent. Pipeline accepts the pre-defined intents and entities and applies all the natural language processing techniques, mentioned in the theoretical background of the work, to perform the process of text-preprocessing (lemmatization, bag of words, pos tagging etc). Text-preprocessing is an important process as it extracts specific text features and thus the system can learn the underlying patterns from the provided examples.

Rasa has a pre-installed model to perform all these above mentioned techniques, which need to be defined in a file, config.yml. This specific file needs to be located in a higher-level folder from the data folder in our computing system. In this file, we must first specify the language, which for this agent will be the English (language: en). Next, we have to define the techniques we want to apply, which already exist in Rasa. The Tokenization process is applied with the "WhitespaceTokenizer", the Bag of Words process with the "CountVectorsFeaturizer", the Entity Recognition process with the "DIETClassifier". Through the use of the term epochs we define the number of times the classifier has to go through the entire training set and with random_seed we define how many times this process will be replicated. Finally, the sequence of procedures applied to a text is not performed randomly but the specific order is defined in the config file. The above procedure is illustrated in the following image:

```
language: en
pipeline:
  - name: WhitespaceTokenizer
  - name: CountVectorsFeaturizer
  - name: DIETClassifier
    epochs: 70
    random_seed: 2
```

Figure 14: Defining config file

6.3.3 Training the Rasa NLU Model

At this stage, we need first to use the Command Prompt (cmd), locate the directory on our computer that we use for our agent and then execute the following command: *python -m rasa* *train nlu*. After this step, we will notice that a "models" folder has been created that contains the results from the above process.

In order to verify the feasibility of this procedure, we execute the following command from cmd: *python -m rasa shell nlu*. After performing this specific task, the related output reveals the following message: *NLU model loaded. Type a message and press enter to parse it. Next message: and the agent is waiting for us to ask him a question*. The following image illustrates the question we queried the agent as well as the related agent's response. As it is shown, the word "hello" that we provided as a message, was grouped in the category "greet" with satisfactory confidence.

```
NLU model loaded. Type a message and press enter to parse it.
Next message:
hello
{
  "text": "hello",
  "intent": {
    "id": 7759222010853559168,
    "name": "greet",
    "confidence": 0.9993851184844971
  },
```

Figure 15: Evaluation of NLU performance

6.3.3.1 Entity Synonyms

It is common for users who wish to search for a specific product from a web page to type it into the search bar in a different way, even using different words/keywords for a specific concept. For example, in the sentences "I would like to buy a phone", "I would like to buy a mobile" and "I would like to buy a smartphone" the user's intention is the same, however it has been expressed differently. Given the lexical richness of most languages (in this case English), it is appropriate to define Entity Synonyms, i.e. synonyms that refer to the same entity and can be recognized by our system. In other words, our target is to extract the appropriate keyword from user's message that will help the agent to understand the user. In order to apply the above procedure, we must include in the config.yml file of Rasa NLU the command that will allow us to use entities in our agent. This can be performed using the "- name: EntitySynonymMapper" command. At the following stage, we need to define the entities. This can be implemented with "## synonym:" as shown in the next image:

```
## synonym:phone
- phone
- phones
- mobile
- mobiles
- smartphone
- smartphones
- cellphone
- cellphones
```

Figure 16: Defining Synonyms

Indicatively, entity synonyms were defined only for the "phone" category. This list can be also enriched with other product categories such as "TV - television", "pc - laptop - computer" etc. At the end of the above procedure, the agent should be trained again in order to incorporate the “new knowledge” into his knowledge based system. This can be implemented with the *python -m rasa train nlu* command, similarly to previous steps. In general, it is desirable, at the end of each step, to repeat the training process to check if every change or addition we made to the agent, worked normally without any errors.

6.3.4 Providing more information to our model

In addition to the above information we have given to our agent, we can train it with frequently asked questions. For example, a user may be interested to know the offers of the online store. For this reason, the conversational agent should be able to answer such questions as trained.

```
## intent: faq/deals
- What kind of deals do you have?

## intent: faq/store_location
- Where is your physical store located?
```

Figure17: Defining frequently asked questions

Clearly the above list can be extended with more examples, such as which payment methods are accepted, what is the product warranty, etc. Below are the indicative answers that the conversational agent can give.

```
utter_faq/deals:
- text: Currently, you can buy 2 phones and get another one for free!

utter_faq/store_location:
- text: We don't have a physical store, we are an online marketplace.
```

Figure18: Defining answers of FAQ

6.4 Rasa Core

In this section we will analyze another component of Rasa, the Rasa Core which is the second part of this system prototype. While the previously analyzed Rasa NLU is related to the understanding of the text entered by the user, the Rasa Core is related to the answers that the agent must give to the user and handle the dialogs between agent and user. Rasa Core does not rely on user-defined dialogs to understand questions and provide answers but uses machine learning techniques to learn conversational patterns from provided data/information. Using this information it predicts then how it should respond for each case, by taking into account the context of the dialogue, such as the previous history and based on other elements we have pre-defined, such as the Policies. In other words, rather than acting as a classic rule-based scheme in the form "if the user says that - then reply that", the agent practically is a trained machine learning model which learn how to handle questions that have not been stated before agent's construction. However, for each case we have to set some rules, called dialogue management. These rules include some questions and answers that the agent must know in advance to know how to proceed.

First of all, regarding the answers that the agent has to provide, certain assumptions need to be defined, i.e. the hard coded messages that the conversational agent should respond when it receives a message from the user. That is, in case the user says "Good evening" to the agent, the agent on its part should return the greeting by saying something appropriate, such as "Good evening, how can I help you?" Rasa allows us in this category to have multiple type responses, such as a message, an image or even a button. On the other hand, in addition to those assumptions, we have additional custom actions, i.e. the agent can call an API or a database to reply to the user's question. For the custom actions process, a python file was created, named actions.py, which defines what Rasa will take as data from users and what exactly it should provide as potential responses. Through custom actions, the agent can understand the content of previous questions and answers given by the user or even which entities have been extracted from the entire discussion. For example, it is defined that in case the user gives one of the following answers, the conversational agent should understand the user's intention and respond appropriately:

```
## intent: goodbye
- Thank you very much!
- See you later!
- Good Night
- bye!
- OK thank you. Goodbye!
- Have a nice day!
- Adios
- Ciao
```

Figure 19: What user might say to the agent

```
utter_goodbye:
- text: Thank you very much for visiting us!
- text: Have a nice day!
- text: bye bye
- text: Ciao!
```

Figure 20: Respond of the agent-based

A very important element that we have to define in Rasa, are the stories, that is, real dialogues that could occur between a person and a salesman. These dialogs are very important for the creation of the agent, as it must be trained on them. It is natural that the more data, i.e. real dialogues we import to the agent for training purposes, it will increases agent's response rates to any user's questions. In fact, the effectiveness of the agent is judged to a large extent by the sample dialogues that we will provide for training. While at first glance, this system seems to be rule-based with if-then clauses, in reality it is not. Machine learning models are applied, because the user, while "talking" with an agent, may ask something unrelated to the discussion, as happens in real dialogues. Thus, the agent should be able to isolate such unnecessary information, respond to it, and then continue the dialogue from the point where that unnecessary question was asked.

However, it is rather an unlikely scenario that we could have defined a-priori all possible questions that a user might ask to the agent. This is due to the lexical richness of natural languages. Each user can express in a unique way his/her intention, such as "I would like to buy a cell phone", "I am interested in a cell phone", "I want a cell phone", "I am looking to buy a cell phone". For the above reason and to avoid cases that Rasa will not be able to respond to, it is very important to mention that Rasa aims to understand the intent of the user only, and not to understand word by word the input it gives. Understanding intent can rely heavily on identifying entities.

6.4.1 Rasa Domain

Domain is the main component of Rasa, i.e. it defines all the elements that the agent should know, how it should react for any question asked, which entities should be taken into account by a question and what should be kept in its memory. We create the domain in a file named domain.yml. At the first level, we define in the domain the intents of the potential user, in the same way we defined them when we designed the Rasa NLU database, such as greet, bye etc. Then, in the next step, we define the entities that interest us. For example, if the dialog is about the stock of a mobile phone, possible words that the user may mention and should be taken as entities are ram, battery, camera, screen, OS. In the next step, we need to define some predefined questions that the agent will ask the user and through the received answers, the related information will be searched in the database. Indicatively, some questions that the agent will ask are listed below:

```
utter_ask_product_ram:
  - text: How much GB ram are you looking for?
utter_ask_product_camera:
  - text: How many MP would you like your camera be?
utter_in_stock:
  - text: You are lucky, we have that phone in stock.
utter_no_stock:
  - We are sorry, don't have that phone available.
```

Figure 21: Indicative responses

6.4.2 Dialogue Management: Policies

As Policy we define the process in which the agent follows certain instructions on how to provide an answer. In addition to the machine learning process mentioned earlier related to the type of responses, we need to define some policies that will help the agent to choose the most appropriate response to the user. One of the main components of Policies is the max history.

The above process checks the history of the dialogue and what has been discussed so far. The agent's administrator is able to determine in advance how many "back-steps" the agent needs to undertake in order to be able to provide an appropriate answer to the next question that it will be asked. As an example, if we desire the agent to take into account only the previous answer of the user, then we will have incomplete information and the agent will not be able to give a representative and appropriate answer.

6.4.2.1 Types of Policies

There are different types/categories of Policies that we can define in Rasa, each one with different characteristics/operation. The first category of Policy is called Memorization, where we do not use any machine learning technique but it is based on the predefined models that we have already set for training in Rasa. Taking into consideration the maximum history that has been set in advance, this method tries to match the current dialogue with Stories that have already been used for agent's training. In a way, it imitates the stories that we have already defined during the creation of the system and tries to predict, based on confidence, the flow of the discussion. A second policy that can be applied to Rasa is called Mapping Policy which incorporates the so-called business logic, i.e. a specific intent is followed by a specific action, regardless of the flow of the discussion. For example, a user can talk to the agent about buying a mobile phone, discuss its features, and at regular intervals he/she may ask the agent a question that is not relevant to the flow of the conversation, like "What time does the store close?". The agent should be able to answer this question and then continue the previous discussion normally.

One policy that uses machine learning to predict the next action is the Keras Policy which utilizes Python's Keras library. This library is based on recurrent neural networks. This method takes into account the previous actions, the intents and entities that have been reported by the user, the maximum history. With Keras Policy we can define some parameters that the agent must fulfil for its operation. First of all, the epochs that define how many times to search for information across the dataset, the Validation Split that determines whether the entire dataset will be utilized, the random seed that determines how to select one random number, a method used mainly during training and finally the batch size according to which the number of examples to be collected in a batch is defined.

In addition to Keras Policy, another method using Machine Learning that can be applied by Rasa is Sklearn Policy, based on Python's Sklearn library. This Policy can use either a logistic regression model or a random forest. Another well-known Rasa Policy is the Transformer Embedding Dialogue (TED) Policy which is based on sequence learning and uses transformers instead of RNN. Its architecture is simpler and faster.

One policy used by almost all conversational agents today is the Fallback Policy. Many times, users ask certain questions in agents, which they are not programmed to answer as they are either not trained in them or because they did not understand the meaning of the question. For this reason, agents must be able to handle such problems and provide answers such as "I couldn't understand you, could you please repeat your question" or even "Could you please rephrase your question?". Therefore, if the agent reaches a point where it has exhausted all the possible valid responses it has acquired through its training and does not have the appropriate confident to provide an answer, we must set a minimum “confident” the agents has to get when recognizing its intention is set. If this number does not meet the criteria, the user's request should be handled as previously stated.

6.4.3 Slots in Rasa

Through the term “Slots” we mean memory, i.e. the data that agent stores and utilizes for the flow of the conversation. For example, data that needs to be memorized by the agent is the user's details, such as name, and even user’s preferences for items that the conversational agent of the specific online marketplace is responsible to deal with. For example, if a user states that he/she wants to buy a device with android software, the agent must store this request in its memory in order to use it and make targeted suggestions to the user. In this way personalization can be fulfilled and we have achieved the so-called customer satisfaction. As mentioned in the theoretical background of Master's Dissertation, Customer Experience is a key factor for a business. By implementing an agent that holds what the user is saying, the agent provides to the user the feeling that the user is not simply talking to a machine but a real human being. In the domain.yml file we already have created, we define the slots, i.e. the categories and the type of each answer we expect, as shown in the following example text.

```
entities:
- category
- ram
- battery
- camera
- budget


slots:
  battery:
    type: text
  budget:
    type: text
  camera:
    type: text
  category:
    type: text
  ram:
    type: text
```

Figure 22: Defining slots

6.4.4 Forms

In case we wish the user to import during the conversation with the agent some data/information, which we will need to use in the future, such as the user's email for sending him/her a newsletter, then such process can be implemented through the Forms offered by Rasa. The data collection through Rasa can be utilized by websites of all types and for different uses. For example, if we have a website with airline tickets, the user must enter the dates and destination for the trip. This data must be stored in Rasa memory so that the agent could use it and draw conclusions. The above procedure is called slot fitting. Rasa also allows us to validate the data entered by the user. For example, if we ask the user to enter a phone number and it consists of missing digits, then we can send a message to the user to correct what he/she has imported. Alternatively, if we ask the user to tell us if he/she wants a mobile phone with android or ios software and the user replies with numbers and not with some text, then the above procedure is also repeated. In order to define Forms we must first define the intents we expect the user to give us, depending on the occasion. For example, if the user is interested in buying a mobile phone from the online marketplace, our agent will ask him/her some questions in order to understand the needs of the user. For this reason, we define certain categories, as described below:

```
utter_ask_survey_form_rating:
- buttons:
  - payload: '1'
    title: '1'
  - payload: '2'
    title: '2'
  - payload: '3'
    title: '3'
  - payload: '4'
    title: '4'
  - payload: '5'
    title: '5'
  text: On a scale of 1-5, how would you rate your conversation today?
utter_ask_survey_form_open_feedback:
- text: Is there any other feedback you can provide?
utter_survey_end:
- text: Thank you for the feedback!
```

Figure 23: Defining Forms

6.4.5 Buttons

In the previous sections, we have described the process by which we expect the user to import the required information, so that the agent can receive it, recognize the user's intention, and then process it in order to provide a response. However, Rasa allows us to have some options, in the form of buttons, according to which certain options will be displayed to the user and he/she in turn will choose the one he/she wishes based on user's preferences. For example, in the agent question "What software do you want your mobile phone to have?", the user will be able, instead of typing what he/she wants, to select one of the buttons that will appear, such as a button with the "Android" option and a button with the "Ios" option. Although the above mentioned option, on the one hand, offers convenience to the user as with the press of a button he/she can select the desirable category without entering the typing process, on the other hand the user gets the feeling that he/she is addressing an automated system without having humanoid characteristics, since this is, after all, the role of modern conversational agents.

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  hi
? Hi there! I'm StavrosBot. How may I help you? (Use arrow keys)
 » 1: Check status of my order (Check status of my order)
   2: Start a return (Start a return)
   3: Check inventory (Check inventory)
   4: Subscribe to product updates (Subscribe to product updates)
   Type out your own message...
```

Figure 24: Example of buttons

The above buttons were defined as a navigation menu of the customers to select one of the categories they want. Users have the option to either select a button or type their own message.

```
utter_greet:
- buttons:
  - payload: Check status of my order
    title: Check status of my order
  - payload: Start a return
    title: Start a return
  - payload: Check inventory
    title: Check inventory
  - payload: Subscribe to product updates
    title: Subscribe to product updates
  text: Hi there! I'm StavrosBot. How may I help you?
```

Figure 25: Defining buttons

6.4.6 Off-Topic Questions

Usually, many times when interacting with an agent, the user may ask the agent something that is not related to the content of the conversation. The agent may also not be programmed to answer an off-topic question. These questions have nothing to do with the content of the agent's online marketplace and may be of a general and vague nature. Alternatively, users may ask agent intentionally to see the agent's reaction. Examples of such questions are like "Where do you live?", "What's the weather like?", "What do you do as a living?", "Tell me your news". Obviously, an agent is neither programmed/trained to answer such questions nor will it ever be able to understand the user's intents and relate them to any of the products it deals with. For this reason, we will have to manually record such questions and provide some ready-made answers. In order to solve the problem of off-topic questions, the solution followed is an If-Else procedure. Obviously, it is not possible to predict all the questions that users may ask and the provided answers should be recorded. The implementation of this task needs to be done by a team of Conversation Designers. As part of this Master's Dissertation, some questions and answers will be recorded, in order to present the process of answering off-topic questions.

```
## intent: off_topic
- what is the meaning of life.
- Fridge Isn't Running
- my tv isn't working
- I want a pizza
- my washing machine isn't working
- what year is it
- I want to order a pizza
- what is the weather today
- what is the weather
- why is the sky blue
```

Figure 26: Defining off-topic questions

6.5 Creating the database

As it has been demonstrated above, in order for the conversational agent to work, we need to enter some data, ie certain question and answer scenarios that may be said by a prospective customer. This conversational agent aims at customer service and one of its main functions is to inform customers about the product stock, to learn about product offers, as well as to get information about the progress of an order. To do all this, there must be a database that will have the products as well as their stock count. This database should be constantly updated, either manually or automatically.

For this reason, an indicative database was created with some products and some of their characteristics, so that the agent can be evaluated later.
To create the database with the product inventory, we create a python file and write the following commands. After that, we run this file and a database file (data.db) is created.

```
import sqlite3
conn = sqlite3.connect('data.db')

c = conn.cursor()

c.execute('''CREATE TABLE inventory
             (GB RAM, MP Camera)''')

inventory = [(2, '12'),
             (2, '14'),
             (3, '17'),
             (4, '19'),
            ]


c.executemany('INSERT INTO inventory VALUES (?,?)', inventory)

conn.commit()

conn.close()
```

Figure27: Creating the database

Once the above process is completed, this file must be placed in the "actions" folder of Rasa, so that it can recognize it. Now, the agent is able, according to the image above, to know that it has 4 devices in stock with specific features (For example, a device with 2 GB of RAM and 12 MP Camera). Of course, the above stock list remains indicative, as we can define specific products (For example "Iphone 12 Pro Max", with specific memory and specific color). To make this possible, we create more tables in the database and respectively in the inventory we add more features.

Another feature provided by the conversational agent of this Master's Dissertation, is to ask about the progress of the order, ie if it is on its way, if it has already been sent, if it will be delayed, etc. In order to do this, the orders and their their situation must be placed in the database. It is becoming clear that for a company that receives hundreds of orders every day, this process can not be done manually. For this reason, the company's system can be connected to the Rasa database, so that every new order is registered and any change is automatically updated. For the purposes of this Master's Dissertation, some indicative customers were added to the previous database with their details and the status of the order, so that it can be searched by the general public.

```
c.execute('''CREATE TABLE orders
             (order_date, order_number, order_email, GB, MP, status)''')

purchases = [('2021-01-02',1111,'customer1@stavrosbot.com','2', 12, 'shipped'),
             ('2021-01-03',1112,'customer2@stavrosbot.com','2', 14, 'order pending'),
             ('2021-01-04',1113,'customer3@stavrosbot.com','4', 19, 'delivered'),
            ]

c.executemany('INSERT INTO orders VALUES (?,?,?,?,?,?)', purchases)
```

Figure 28: Creating the database

6.6 Demonstration

After we have applied all the above, the conversational agent works normally for the functions we have programmed. The agent demonstration can be done via Rasa Shell from the Command Prompt. However, Rasa can also be linked to a website to provide a graphical interface if a chatbot interface already exists, or to be linked to an application such as Facebook Messenger, Telegram, etc. However, this is not possible in a Master's Dissertation as we do not possess any ecommerce store.

To start the agent, through the cmd of our computer, we locate the folder of the agent and type "rasa shell" to start working. The pictures below show its effectiveness, as well as the work environment that has been evaluated by the volunteers in the next chapter.

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->
```

Figure 29: Demonstration

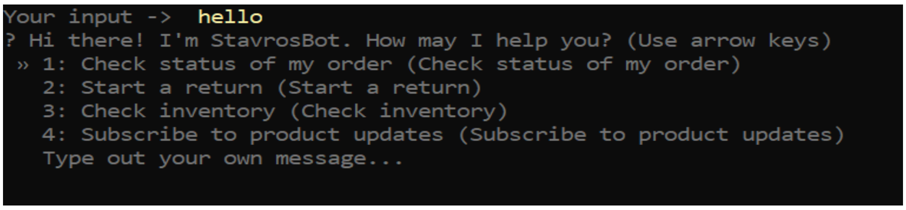


Figure 30: Demonstration

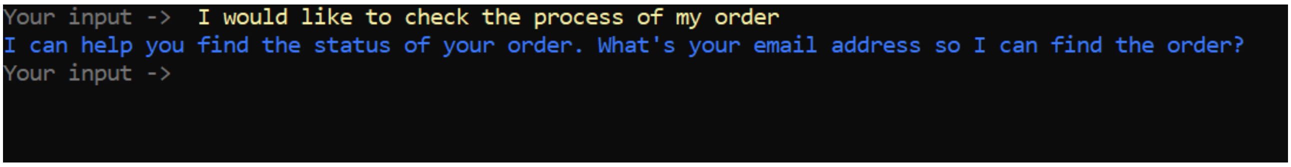


Figure 31: Demonstration

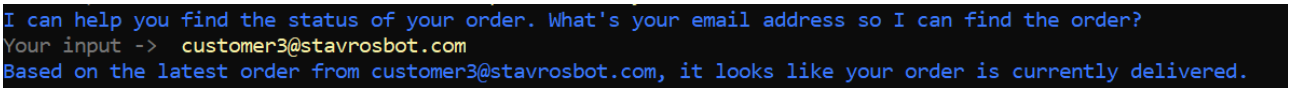


Figure 32: Demonstration

Chapter 7
Evaluation

For the evaluation of this conversational agent, 60 volunteers, aged 24 to 45, participated and were asked to discuss with the agent and then complete a questionnaire. Before the participation of each volunteer, the environment of the agent has been explained. The conversational agent was installed on a local workplace computer as due to lack of resources it was impossible to rent a server which would be widely available to the general public. Most companies that incorporate such systems have high-cost servers for running similar agents, which is not possible in a Master's Dissertation context. In the questionnaire given for completion, there was the following description.

"This questionnaire was conducted in the context of the Master's Dissertation of Stavros Leonidis for the MSc Cognitive Systems, offered by Open University of Cyprus. Participation in this research is anonymous and the results will be collected exclusively to draw conclusions about the conversational agent that has been created. For your participation, imagine that you are in an e-commerce store with technology products. You are asked to interact with the agent and ask it questions that you would ask an employee in a retail store."

Volunteers were asked to talk to the conversational agent and discuss with it about technology products. It has become clear from the outset that, since it is an individual endeavor, it has not been programmed by a team of conversational designers to anticipate all the questions or answers that will be conducted. They were asked to be objective in their evaluation of the agent and to answer the questionnaire with complete honesty.

The first question was whether they have previously reached an agent and the results are presented in the table below. It is understood that most participants have not used an agent in the past to assist them with online shopping. It is a fact that most Greek retail websites do not have agents, so the result from the above question is to be expected.

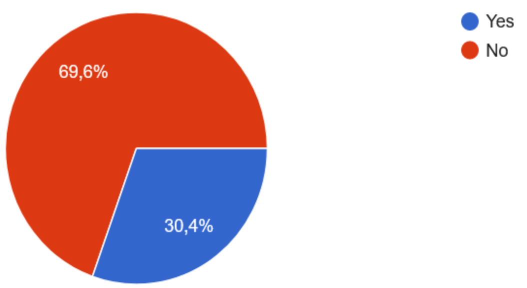
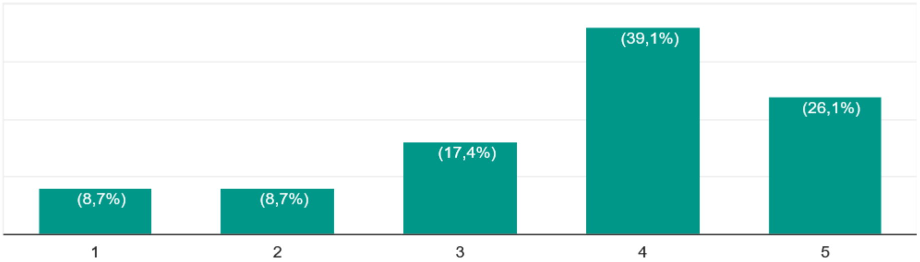


Figure 33: Question No. 1

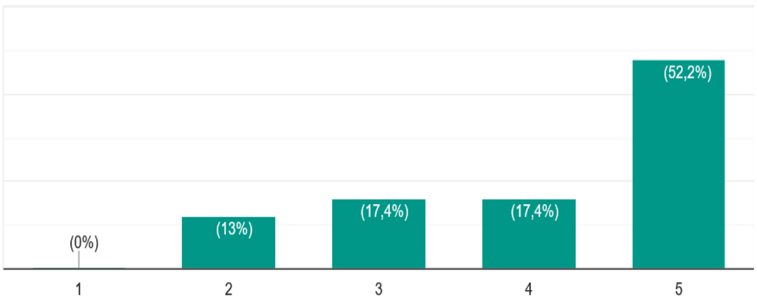
Candidates were then asked if they consider agents to be credible, on a scale of 1 to 5. Most candidates, although they had not previously contacted an agent other than this one of the Master's Dissertation, answered positive in the above question with most giving a score of 4



and the minority giving low scores, 1 to 2.

Figure 34: Question No. 2

The next question was whether candidates consider conversational agents to be an advantage for a business. Almost all the answers were positive with most of the participants, namely 52.2% giving a grade of 5. The above finding confirms what was said in the theoretical background of the Master's Dissertation on the advantages that a company acquires by applying such technologies. Conversational agents are a quick way to get in touch with a business without the hassle of waiting, whether it's a call center or waiting for a reply to an

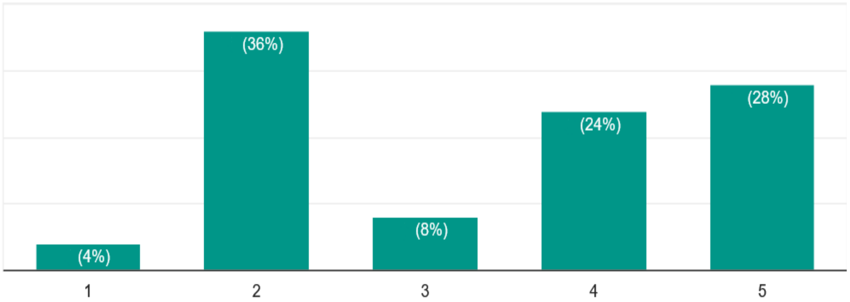


email.

Figure 35: Question No. 3

We then asked the volunteers if an agent could serve a customer just like an employee. 36% answered with 2 showing that it is not possible to achieve such a thing while, as shown below in the graph, there were cases of volunteers who answered positively to the above question. The issue of equal customer service by an agent and an employee is questionable.

Conversational agents have not been applied in market for years and specifically in Greece, where the present research was conducted, therefore people are not yet accustomed to this



logic and mentality, being served by a machine.

Figure 36: Question No. 4

The next question was mainly about the trust of the clients towards agents. Specifically, it was asked if customers would trust personal information to agents, such as a password. Most of the answers, as shown below, were 1 to 3 and in this way the distrust of the clients towards the agent was shown. It makes some sense for customers to express distrust of a computer, as they may not know how their data will be used by the agent.

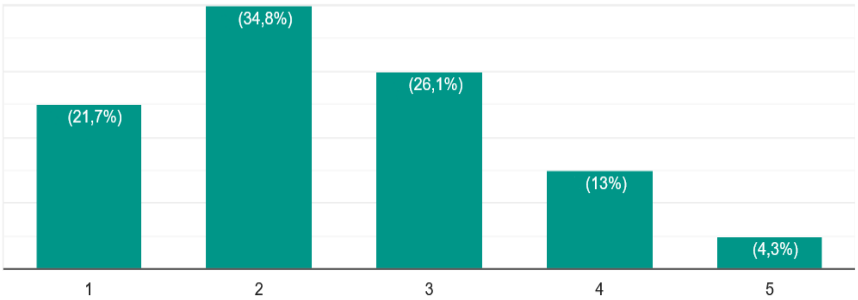


Figure 37: Question No. 5

The following question was about the agent of this Master's Dissertation. Volunteers were asked to rate the agent and answer on a scale of 1 to 5 whether they were satisfied with its performance. There was no clear answer, as the answers were mixed. In this case, it seems that some were very satisfied, some less and some not at all. It has become clear from the beginning of the investigation that this agent is merely a demonstration of an agent implemented by online marketplaces. Logic says that the negative answers come mainly from people who had used an agent in the past, 30.4% of the volunteers, and had different expectations from its operation. However, there were volunteers who were positive about the agent, for what it has been trained to do so far.

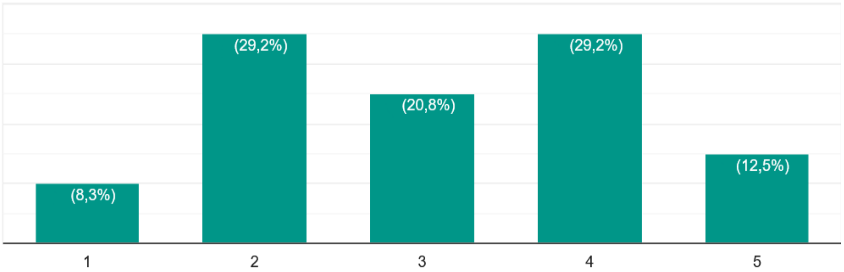


Figure 38: Question No. 6

Candidates were then asked to indicate whether the particular agent had served them, such as a store employee. These answers are in line with the answer given to a previous question concerning customer satisfaction with agents in general. Therefore, the results from this question given below were expected.

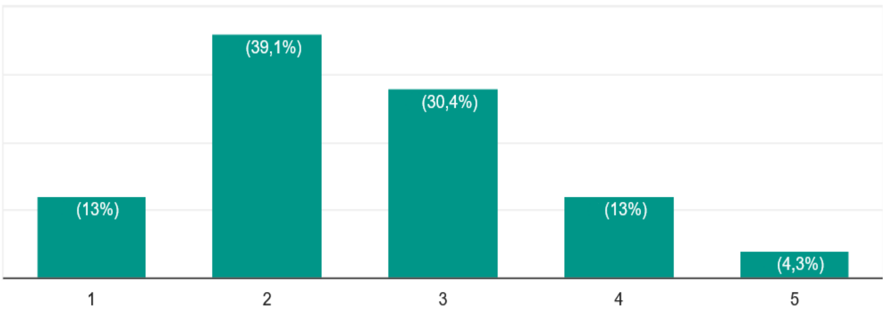


Figure 39: Question No. 7

The next question was about the intentions of the candidates. They were asked to rate whether the agent could understand the intentions of the candidates and give the appropriate answers. 39.1% of the candidates answered with 3, which is judged as an expected result as it has not been trained with a huge dataset and are not able to understand all the inputs from the users.

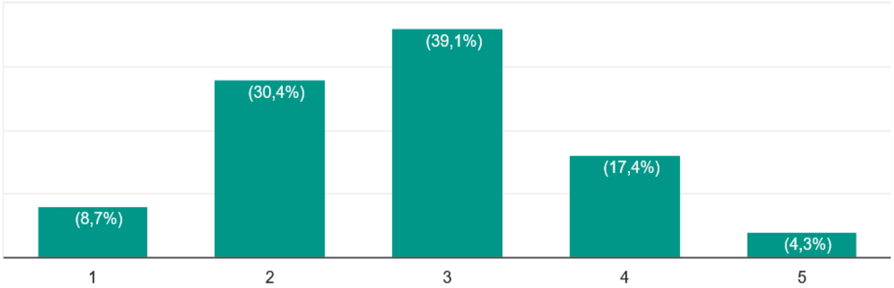


Figure 40: Question No. 8

We then asked the candidates if the conversation they had with the agent had a flow. To this question, the answers were mixed without receiving a clear answer in the opinion of the candidates. As can be seen in the chart below, some candidates felt that there was a normal flow in the discussion they had with the agent, while others apparently were not so pleased.

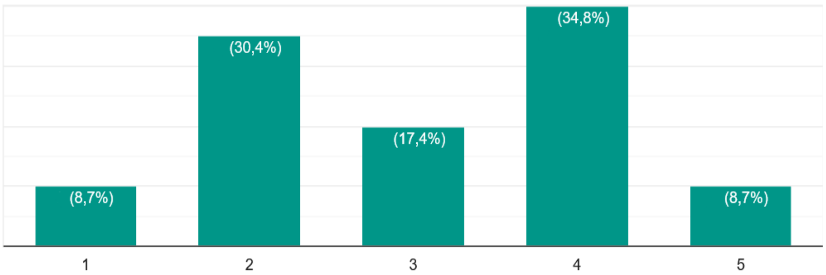


Figure 41: Question No. 9

The last question in the questionnaire was asking the candidates if after this experience, they would use a conversational agent in the future. Most of the candidates seemed positive with the agent and would like to have similar experience in the future.

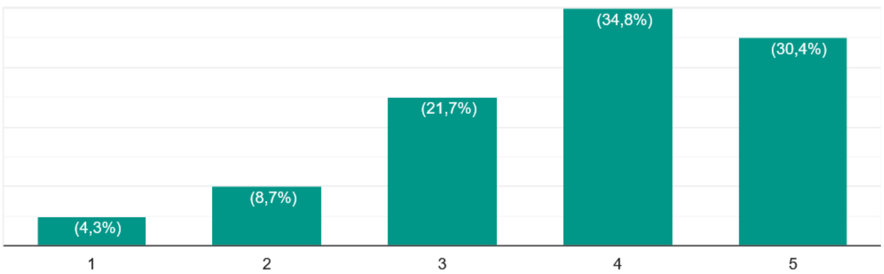


Figure 42: Question No. 10

Chapter 8
Conclusion

In the context of this Master's Dissertation, conversational agents and their contribution to e-commerce have been investigated. Through the first chapters, the theoretical background of computer science has been presented, while the development of conversational agents and their potential advantages to community have been addressed. Furthermore, the natural language processing techniques used by most modern conversational agents were explored, and a state-of-the-art review of the most well-known - commercial conversational agents currently available on the internet was also presented. As discussed earlier in the theoretical part of this Master's Dissertation, conversational agents constitute currently a very important element that e-commerce stores need to have for customer service. They incorporate features and technologies that could provide several benefits to a business. Some of these include the increase in Customer Experience, as the customer will be served in real time without waiting to talk to an employee as well as the increase in business profits, as a user will not have to rely on a customer service department, as this process will be done automatically. However, in the framework of a Master's Dissertation, it is not possible to fully implement a conversational agent, similar to those developed/used by the retail companies and have been integrated in their websites. The lack of resources is the main reason for this issue. Large companies that integrate conversational agents into their systems maintain high-cost servers on a monthly basis, which are able to receive and process large volumes of data with unlimited traffic. The proposed in this dissertation conversational agent was created with the aim to show the philosophy behind this system by demonstrating the way and techniques it requires for an efficient operation. The purpose of this agent is to operate locally and guide users to supply a product according to their provided options/requests. It is a fact that the effectiveness of the conversational agent depends to a great extent on the training stage that has been carried out. Its ability to recognize a large number of questions from users is due in large part, to the effort made by Conversational Designers to provide it with an adequate dialogue templates. Its ability also depends on the dataset that has been trained to be able to identify any intent of the user, even if it is not included in the templates given to it.

This development of this conversational agent obviously can be improved and this could be considered as a future work. First of all, its knowledge seems to be limited in answering users' simple questions only as it is not able to serve users to place new orders. Practically, it is considered as an informative agent for the orders and the availability of products. At the same time, another improvement that could be considered is the potential link of the customer with a customer service employee, when it is not possible for a user to be served by the agent. Finally, a very important addition would be the speech support from the agent, i.e. a voicebot, a capability that is fully supported by Rasa.

Chapter 9
References

Annika, Salmi. (2020). Artificial Intelligence Enabled Solutions in Marketing: Case Ekokompassi

Bezboruah, Tulshi & Bora, Abhijit. (2020). Artificial intelligence: The technology, challenges and applications. 10.14738/tmlai.85.8956.

Biswas, Manisha. (2018). Beginning AI Bot Frameworks: Getting Started with Bot Development. 10.1007/978-1-4842-3754-0.

Brandao, Pedro. (2018). HISTORY OF INFORMATICS: THE BEGINNING OF COMMERCIAL COMPUTERS. International Journal of Current Research. 10. 70216-70218.

Çelik, Özer. (2018). A Research on Machine Learning Methods and Its Applications. 10.31681/jetol.457046.

Dale, Robert. (2020). Natural language generation: The commercial state of the art in 2020. Natural Language Engineering. 26. 481-487. 10.1017/S135132492000025X.

Duijst, Daniëlle. (2017). Can we Improve the User Experience of Chatbots with Personalisation?. 10.13140/RG.2.2.36112.92165.

Følstad, Asbjørn & Skjuve, Marita & Brandtzaeg, Petter. (2019). Different Chatbots for Different Purposes: Towards a Typology of Chatbots to Understand Interaction Design. 10.1007/978-3-030-17705-8_13.

French, Robert. (2000). The Turing Test: The first 50 years. Trends in Cognitive Sciences. 4. 115-122. 10.1016/S1364-6613(00)01453-4.

Gentile, Chiara & Spiller, Nicola & Noci, Giuliano. (2007). How to Sustain the Customer Experience:: An Overview of Experience Components that Co-create Value With the Customer. European Management Journal. 25. 395-410. 10.1016/j.emj.2007.08.005.

Hoffman, Novak (2009). Flow online: Lessons learned and future prospects. Journal of Interactive Marketing, 23(1), 23–34

Hung, Victor & Elvir, Miguel & Gonzalez, Avelino & DeMara, Ronald. (2009). Towards a Method For Evaluating Naturalness in Conversational Dialog Systems. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. 1236 - 1241. 10.1109/ICSMC.2009.5345904.

Hussain, Shafquat & Sianaki, Omid & Ababneh, Nedal. (2019). A Survey on Conversational Agents/Chatbots Classification and Design Techniques. 10.1007/978-3-030-15035-8_93.

Klaus, Phil & Maklan, Stan. (2013). Towards a Better Measure of Customer Experience.

International Journal of Market Research. 55. 227-246. 10.2501/IJMR-2013-021.

Kuligowska, Karolina. (2015). Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents. Professionals Center for Business Research. 2. 1-16. 10.18483/PCBR.22.

Li, Chenzhuoer & Pan, Runjie & Xin, Huiyu & Deng, Zhiwen. (2020). Research on Artificial Intelligence Customer Service on Consumer Attitude and Its Impact during Online Shopping. Journal of Physics: Conference Series. 1575. 012192. 10.1088/1742-6596/1575/1/012192.

Mohsienuddin, Sikender & Syed, Habeebullah Hussaini. (2020). ARTIFICIAL INTELLIGENCE IN INFORMATION TECHNOLOGY. SSRN Electronic Journal. 7. 168-176.

Novak, Hoffman (1997). Measuring the flow experience among web users.

Novak, Hoffman, Duhachek (2003). The Influence of Goal-Directed and Experiential Activities on Online Flow Activities. Journal of The American Academy of Dermatology - J AMER ACAD DERMATOL. 13. 3-16. 10.1207/S15327663JCP13-1&2_01.

Novak, Hoffman, Yung (2000). Measuring the Customer Experience in Online Environments: A Structural Modeling Approach. Marketing Science. 19. 22-42. 10.1287/mksc.19.1.22.15184.
Powers, David. (2002). The Total Turing Test and the Loebner Prize. 10.3115/1603899.1603947.

Rose, Clark, Samouel, Hair. (2012). Online Customer Experience in e-Retailing: An empirical model of Antecedents and Outcomes. Journal of Retailing. 88. 308–322. 10.1016/j.jretai.2012.03.001.

Sharma, Rakesh. (2020). An Analytical Study and Review of open source Chatbot framework, Rasa. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS060723.

Svenningsson, Nina & Faraon, Montathar. (2019). Artificial intelligence in conversational agents: A study of factors related to perceived humanness in chatbots. 10.1145/3375959.3375973.

Tamrakar, Rohit & Wani, Niraj. (2021). Design and Development of CHATBOT: A Review.

Verhoef, Lemon, Parasuraman, Roggeveen, Tsiros, Schlesinger. (2009). Customer Experience Creation: Determinants, Dynamics and Management Strategies. Journal of Retailing. 85. 31-41. 10.1016/j.jretai.2008.11.001.

Weizenbaum, Joseph. (1966). "ELIZA—a computer program for the study of natural language communication between man and machine." Commun. ACM 9: 36-45.

Wong, Wilson & Thangarajah, John & Padgham, Lin. (2012). Flexible Conversation Management Using a BDI Agent Approach. 7502. 464-470. 10.1007/978-3-642-33197-8_48.

Zumstein, Darius & Hundertmark, Sophie. (2017). Chatbots – An Interactive Technology for Personalized Communication, Transactions and Services. IADIS International Journal on WWW/Internet. 15. 96-109.