

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN  
INGENIERÍA DE TELECOMUNICACIÓN**

**TRABAJO FIN DE MASTER**

**Desing and development of chatbot trained with machine  
learning models**

**IGNACIO CERVANTES VILLALÓN**

**2020**



## TRABAJO DE FIN DE MASTER

**Título:** Diseño e implementación de un chatbot entrenado mediante modelos de machine learning

**Título (inglés):** Desing and development of chatbot trained with machine learning models

**Autor:** Ignacio Cervantes Villalón

**Tutor:** Carlos Ángel Iglesias Fernández

**Ponente:** PONENTE

**Departamento:** Departamento de Ingeniería de Sistemas Telemáticos

## MIEMBROS DEL TRIBUNAL CALIFICADOR

**Presidente:** —

**Vocal:** —

**Secretario:** —

**Suplente:** —

**FECHA DE LECTURA:**

**CALIFICACIÓN:**



**UNIVERSIDAD POLITÉCNICA DE MADRID**

ESCUELA TÉCNICA SUPERIOR DE  
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos  
Grupo de Sistemas Inteligentes



TRABAJO DE FIN DE MASTER

Desing and development of chatbot trained with machine  
learning models

MES 20XX



# Resumen

---

En los tiempos que corren, la interacción con sistemas automáticos es una parte fundamental a la hora de realizar tareas cotidianas en sistemas informáticos. Realizar una gestión sobre un pedido en una tienda online o solicitar asistencia técnica son algunas de las aplicaciones que suelen recaer en automatismos para ofrecer una respuesta rápida, acotada y certera. En este contexto, los llamados Chatbots, consistentes en una inteligencia artificial capaz de simular el comportamiento conversacional de una persona, permiten ofrecer una interfaz amigable para automatizar tareas. El objetivo de este TFM es el de desarrollar un caso de estudio de la aplicación de un chatbot para llevar a cabo el servicio de atención al consumidor. El objetivo de este TFM es estudiar el panorama de los chatbot en la actualidad, mostrando su evolución desde sus inicios hasta el día de hoy comparando sus características y mostrar un caso práctico implantando uno de ellos. Para profundizar en el estudio, se va a desarrollar un caso práctico de uso del chatbot de código libre RASA, haciendo uso de técnicas de machine learning para entrenar sus modelos y comparando los resultados obtenidos mediante la conversación con el robot. Se pretende con ello ver las posibilidades de evolución de un chatbot al involucrar aprendizaje en el modelo de conversación del robot, y comparar los resultados obtenidos con los distintos algoritmos utilizados. Para ello se utilizará Jupyter Notebook y Python.

**Palabras clave:** Chatbot, RASA, Machine Learning, Python.





# Abstract

---

Nowadays, interaction with automatic systems is a fundamental part in order to perform day-to-day tasks in informatic systems. Managing an online shop order or asking for technical assistance are some of the applications where automatisms can offer a fast and precise response. In this context, the so-called Chatbots, that consist in an AI capable of simulating an human-like conversation, can offer a friendly user interface to automatize these tasks. This TFM objective is to develop a case of study of a chatbot applicated to a customer service. In order to do this, a practical case of a open-source code chatbot called RASA is proposed, using machine learning techniques to train its models, comparing the results obtained in a conversation with the bot. The objective is to check the possibilities of evolution that chatbot when machine learning in the conversational model is involved, and compare the results of different machine learning algorithms. The development will be done using RASA as the chatbot where we are going to develop the use case, Jupyter Notebook to make human-friendly code and keep the results organized, Python 3.8 to run the algorithms, notebooks and RASA.

**Keywords:** Chatbot, RASA, Machine Learning, Python



# Agradecimientos

---

A Gauss



# Contents

---

<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>Agradecimientos</b>	<b>XI</b>
<b>Contents</b>	<b>XIII</b>
<b>List of Figures</b>	<b>XV</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Project goals . . . . .	2
1.3 Structure of this document . . . . .	3
<b>2 State of Art</b>	<b>5</b>
2.1 Sentiment Analysis . . . . .	6
2.2 Neural Networks . . . . .	7
<b>3 Enabling Technologies</b>	<b>9</b>
3.1 Introduction . . . . .	10
3.2 Cognitive computing . . . . .	10
3.2.1 Cognitive computing properties . . . . .	11
3.2.2 Use Case of Cognitive Systems . . . . .	11
3.3 Chatbots . . . . .	13

3.3.1	Chatbots Design Techniques . . . . .	14
3.3.2	Chatbot Frameworks . . . . .	15
3.3.2.1	IBM Watson Conversation . . . . .	15
3.3.2.2	Dialogflow . . . . .	16
3.3.2.3	RASA . . . . .	17
3.4	Conclusions . . . . .	19
<b>4</b>	<b>Architecture and Methodology</b>	<b>21</b>
4.1	Methodology . . . . .	22
4.2	Architecture . . . . .	23
<b>5</b>	<b>Case study</b>	<b>25</b>
5.1	Techniques . . . . .	26
5.2	Datasets . . . . .	26
5.3	Results . . . . .	26
<b>6</b>	<b>Conclusions</b>	<b>27</b>
6.1	Achieved Goals . . . . .	28
6.2	Conclusion . . . . .	29
<b>A</b>	<b>Example</b>	<b>31</b>
	<b>Bibliography</b>	<b>32</b>

## List of Figures

---

3.1	Person using Microsoft Hololens [9] . . . . .	12
3.2	Conversation between a person and ELIZA [17] . . . . .	13
3.3	Components of a chatbot [10] . . . . .	15
3.4	Watson Assistant solution for customer service [14] . . . . .	16
3.5	Dialogflow architecture overview [5] . . . . .	17
3.6	Incoming message processing [5] . . . . .	19





# Introduction

---

## 1.1 Context

Nowadays, it is common to interact with business to request services such as technical attention or responses with advice using their products. Interacting in social networks such as Twitter or Facebook is a familiar way to users to get in touch with the support they need. For example, telecommunication service providers such as Movistar or Orange can be reached quickly by writing a message in a social media, instead of a complicated and usually longer telephonic conversation. In these situations, getting a fast response can mean the difference between a happy customer or losing it.

In this scenario, it is desirable to automatize as much as possible to offer a quick response and save costs. To do this and offer a person-like behavior, a common solution is deploying a chatbot. This way, the service provider can give the user a way of getting the information needed in a friendly environment.

## 1.2 Project goals

Describe the different goals of the project

## 1.3 Structure of this document

The remaining of this document is structured as follows:

*Chapter 2* ... (Chapter 1 is the Introduction)



## CHAPTER 2

# State of Art

---

*Introduction of the state of art*

## **2.1 Sentiment Analysis**

## 2.2 Neural Networks





## Enabling Technologies

---

*This chapter offers a brief review of the main technologies that have made possible this project. For the realisation of this TFM, Python is the main programming language. To make it easier to track, Jupyter Notebook is used. Jupyter allows the user to execute the code in cells and see the results for each one, that makes simpler reading the code. To develop the chatbot, the open-source framework RASA is used. RASA allows to develop a chatbot, train its models with machine learning and improve the performance through analysing conversation with RASA X. The following sections explain the fundamentals of these packages and others that are used in the consecution of this project.*

### 3.1 Introduction

To develop this project, many technologies are involved that need to be understood to. In this chapter, we are going to analyze and explain some of them to offer a wider vision of how the project is achieved.

First of all, we are going to explain the technology behind the applications that are used in this project, Cognitive computing. We will deepen in the definition and explain some of its characteristics, as well as list some of their applications.

Secondly, we will analyse the conversational bots by its characteristics, and offer a wide vision of the different applications that are being used now. The chatbots that we are going to cover are IBM Watson, Google Dialogflow and RASA.

Finally, we are going to give conclusions about this chapter.

### 3.2 Cognitive computing

As artificial intelligence and new ways of programming grow, the principal companies have found new advantages in their usage. IBM, Microsoft or Facebook are making great investment in this technologies to make their companies more relevant, and take a leap to cognitive applications. "Cognitive computing" is a term that has been in the sector for years with different definitions. As stated by the Cognitive Computing Consortium[2], a cross disciplinary group of experts, cognitive computing may be defined as:

*"Cognitive computing makes a new class of problems computable. It addresses complex situations that are characterized by ambiguity and uncertainty; in other words it handles human kinds of problems. In these dynamic, information-rich, and shifting situations, data tends to change frequently, and it is often conflicting. The goals of users evolve as they learn more and redefine their objectives. To respond to the fluid nature of users' understanding of their problems, the cognitive computing system offers a synthesis not just of information sources but of influences, contexts, and insights. To do this, systems often need to weigh conflicting evidence and suggest an answer that is "best" rather than "right"*".

Unlike current computing paradigms, cognitive computing model infers some data that is out of scope for basic computing. This way, cognitive computing makes context computable, taking account and using some state-of-the-art technologies like machine learning, natural language processing or feature engineering. In other words, as the Cognitive Computing Consortium states, cognitive computing applications "suggest an answer that is "best"

rather than "right".

To sum up, cognitive computing aims to endow computer systems with the faculties of knowing, thinking and feeling[13] to close the gap between people and the informatic environment. For this reason, it is the basis of technologies like virtual assistants or chatbots. However, it is thought that new applications of cognitive computing will appear in other fields of knowledge.

### 3.2.1 Cognitive computing properties

As the Cognitive Computing Consortium states, there are some features that a cognitive system has to fulfil:

- **Adaptive:** They must be capable of learning as information and goals evolve. They must resolve ambiguity and tolerate unpredictability. Also, they must be engineered to feed on dynamic data in real time, or near real time.
- **Interactive:** A cognitive system must be easy to interact with, so the users can define their needs and requirements in a simple way. They may also be capable of interact with other actors and devices as other users, Cloud services or online applications.
- **Iterative and stateful:** They must be capable to aid the user in defining a problem by asking questions or finding additional source input if a problem statement is ambiguous or incomplete. They must take in consideration previous interactions in a process and return information that is suitable for the specific application at that point in time
- **Contextual:** They must be capable of understand and extract contextual elements of the information, such as meaning, syntax or time. Also, they may be able to obtain information from various sources, structured or not, as well as sensory inputs if the system support them.

### 3.2.2 Use Case of Cognitive Systems

As cognitive systems advance, more and more applications that uses them appear and their relevance rises. Some of the most relevant use cases are:

- **Virtual Private Assistant (VPA):** VPAs[15] are *Artificial Intelligence* applications that assist the user utilizing their devices. They act as an "human-like" interface

to perform some tasks, launched by the user by giving orders to the assistant via voice or text.

Most VPAs are present in several devices and offer services such as access to device functions as making a phone call or writing a message, appointment management or domotic control, and they are linked to personal accounts to take in consideration user's profile and preferences. Some of the most popular are Apple's Siri, Google Assistant, Microsoft's Cortana or Amazon Alexa.

- **Augmented reality:** *Augmented Reality* (AR) technology is a variation of virtual environments where the user is able to see the real world with virtual objects composited in it. Therefore, AR "supplements reality, rather than replacing it"[11].

AR applications take information from cameras to put into the captured images 3D models or layers of information that are reactive to the environment and are capable of adapt to it. These applications can run on various devices, being the most typical smartphones or *Head Mounted Devices (HMD)* such as Microsoft Hololens[8].



Figure 3.1: Person using Microsoft Hololens [9]

Some of the applications of this technology are engineering, 3D design or videogames such as Pokémon GO.

- **Face recognition:** Face recognition allows to use images capted by a camera to identify a person face. This systems recognize patrons in faces through AI to identify univocally a person.

This technology is widely used in biometric security authentication systems, such as Apple's Face ID, or in social networks like Facebook to tag users automatically by

identifying their faces in photographs uploaded by their contacts.

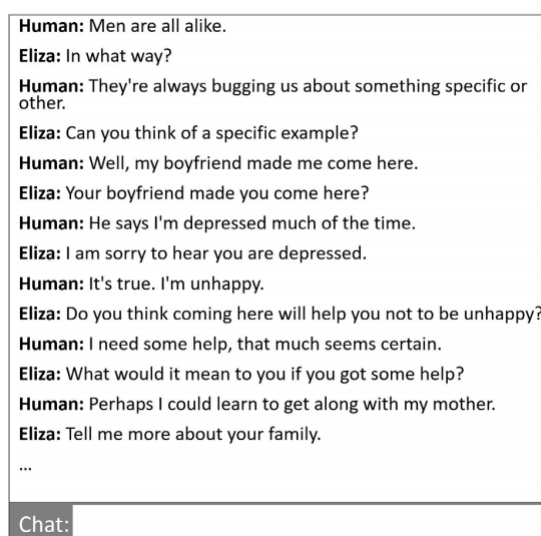
### 3.3 Chatbots

Since the beginning of domestic computers, there was always an interest in giving the user a friendly interface to use the computational resources. Investing in a clear user interface or designing an attractive and simple website makes the user want to use these resources. As the number of people that uses computers rose, it turned to be a necessity that may decide if an application success or fails. In these context, giving the user a natural language interface to interact with is desirable.

As speech is one of the main forms of communication between humans, the researchers aimed to inspire from it. Speech interaction is gaining interest in the past few years, as can be seen from developments from Google, Apple or IBM.

*Chatterbots* (or, from now, *chatbots*) are applications designed to imitate human-like conversation. From the simplest models to modern AI-powered, the goal is to offer a more friendly way to the user to interact with a digital system.

The first recognized chatbot was *ELIZA*, a program created by Joseph Weizenbam at MIT in 1966, that communicated with humans based on hand-crafted scripts. The conversation scope was limited as it was designed using pattern recognition, but enough to keep a realistic conversation, as can be seen in the figure below.



**Human:** Men are all alike.  
**Eliza:** In what way?  
**Human:** They're always bugging us about something specific or other.  
**Eliza:** Can you think of a specific example?  
**Human:** Well, my boyfriend made me come here.  
**Eliza:** Your boyfriend made you come here?  
**Human:** He says I'm depressed much of the time.  
**Eliza:** I am sorry to hear you are depressed.  
**Human:** It's true. I'm unhappy.  
**Eliza:** Do you think coming here will help you not to be unhappy?  
**Human:** I need some help, that much seems certain.  
**Eliza:** What would it mean to you if you got some help?  
**Human:** Perhaps I could learn to get along with my mother.  
**Eliza:** Tell me more about your family.  
...  
Chat:

Figure 3.2: Conversation between a person and ELIZA [17]

Another milestone in chatbot history was Parry, a chatbot developed by Colby designed to have a paranoid behavior in conversation. Parry was the first chatbot that passed the Turing test, that means that it showed a indistinguishable behaviour from a human[16]. It was also ruled-based as ELIZA, but Parry had a better language understanding capabilities and models that allowed it to mimic human emotions. For example, it was capable of showing hostility if the anger level in conversation was high.

However, the more recent and modern approaches of chatbots are not longer based on pattern recognition. The chatbots that are listed in the next sections are based in *Natural Language Processing* (NLP) and machine-learning techniques.

### 3.3.1 Chatbots Design Techniques

To be able to give suitable responses to the user inquiries, some programming skills and techniques must be used to program a chatbot. In this section we are going to briefly explain them, as well as understanding the different parts of a chatbot.

As chatbots evolve into more realistic conversational partners, the complexity in programming them also multiplies. For this reason, designing a chatbot require identifying its main parts. A Chatbot can be classified into three main parts[10]:

- **Responder:** This part acts as the interface between the bot logic and the user. Its goal is to pass the data from the user to the next step and controlling the inputs and outputs of the system.
- **Classifier:** This layer connects the responder and the graphmaster, and is able to filter and normalise the inputs from the user to pass it to the graphmaster. It also processes the output of the graphmaster if needed.
- **Graphmaster:** This part is in charge of the logic of the chatbot. It keeps the storage and performs the operations needed to determinate the response (i.e. pattern matching)

To design a chatbot, some programming techniques and skills are needed. Some of the most important are:

- **Parsing:** Analysing the input text received from the user and manipulate it (i.e. morphemes recognition) with NLP functions.
- **Pattern matching:** Detecting patterns in the user messages to catch the meaning or the sentiment of them, and respond in consonance.

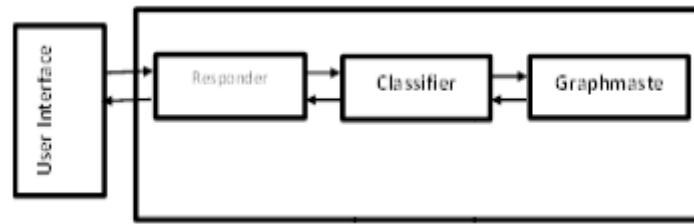


Figure 3.3: Components of a chatbot [10]

- **Markov Chain:** It is a stochastic model widely used in chatbots to give responses taking account of previous responses, which means that the next state probability is conditioned by the previous states.

### 3.3.2 Chatbot Frameworks

The number of chatbot frameworks has been increasing since a few years ago, at the same time as they evolved into more interactive and functional tools. The evolution of webs also allowed to embed chatbots into websites to give the visitor a way to interact with the company without having a person 24/7. As many companies noticed how the inclusion of a chatbot made an impact in their economy, the interest for different options rose.

There are many benefits of implementing a chatbot for a company in its website. Having a chatbot usually offers an instantaneous responsive option for an user, reducing support cost. Also, it makes a company easier to operate worldwide, as a chatbot is always operative instead of having to adapt to local hour. In terms of savings, about of 30% of costs in support can be reduced[1].

Chatbots can be found in many different applications and forms. In this section we are going to introduce some of the most important frameworks and chatbots used in the present.

#### 3.3.2.1 IBM Watson Conversation

IBM Watson is a conversation tool, initially designed as a question answering system based on natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning[3]. It can be deployed in cloud services such as IBM Cloud, Amazon Web Services, Azure or Google Cloud, and the service can be developed using different languages SDKs, such as Java or Python.

As a chatbot, some of the features that Watson offers is to interact with the bot through voice and text. Once the message is sent to the assistant, the content is analysed through the system. As a differential point with other assistants, Watson decides when to search for an answer from a knowledge base, when to ask for clarity and when to escalate the request to a human to resolve it. Also, as it can be deployed in the cloud service that the user decides, the sensible data and conversations are only stored in it, and not collected by the chatbot provider. This allows to train the bot with that data, and preserve the privacy of the users.

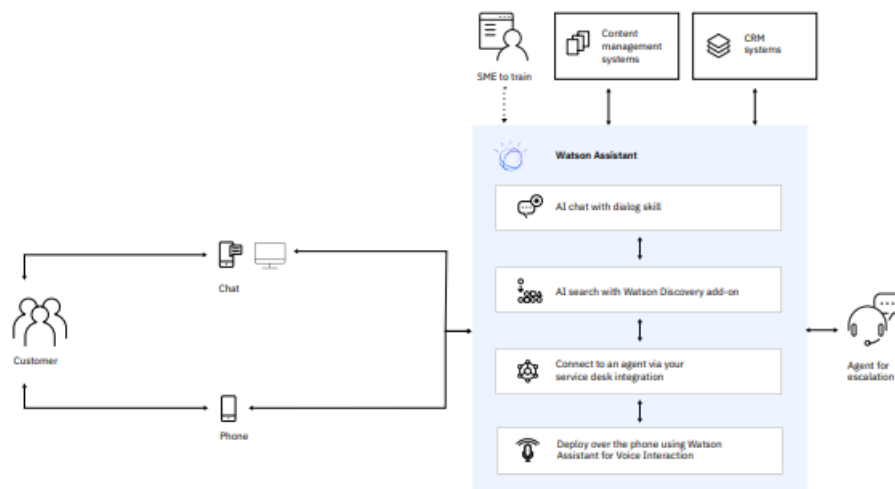


Figure 3.4: Watson Assistant solution for customer service [14]

### 3.3.2.2 Dialogflow

Dialogflow is a conversational tool developed by Google and deployed by Google Cloud. It is based on *Natural Language Processing* and machine learning techniques, and it is one of the most important conversational AI in the market. Some of their clients are big companies such as Ticketmaster or Domino's Pizza, which uses the chatbot to make orders[4].

Google describes Dialogflow as "an end-to-end, build-once deploy-everywhere suite for creating conversational interfaces"[5]. Some of its strengths are

Dialogflow functionality is achieved by different parts that conform the full stack:

- **Agent:** An agent is a virtual entity that handles conversation with the end-user. It interacts in natural language, via text or audio, and is trained to handle expected conversations with the user.
- **Intents:** Every user intention is categorized in "intents". The agent is trained to



recognise the differences between them, extract information from them and perform an action as a response. An intent contains *training phrases*, that are example phrases that the user might send, *actions* that triggers certain logic in the system, *parameters* extracted from the user message that can be used through conversation and *responses*, which continues the conversation with the user.

- **Intents:** Dictates how data obtained from the user is extracted and managed
- **Contexts:** Interpretation by Dialogflow for the context of the conversation. This allows to match an intent to some specific context to make Dialogflow more likely to match the next intents related to that context, just as human communication works.
- **Fulfillment:** As Dialogflow can integrate other service in order to give a response, fulfillments functions connecting an intent with a call to a service defined. For example, it can match a question about the weather with a search in a database with a weather history.

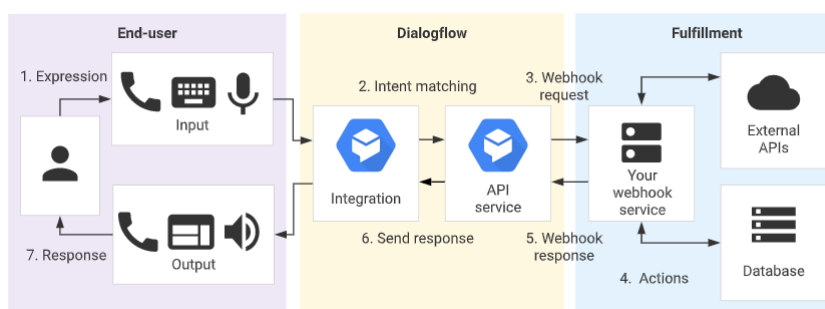


Figure 3.5: Dialogflow architecture overview [5]

### 3.3.2.3 RASA

RASA is a conversational AI framework for building contextual assistants. Unlike IBM Watson or Google Dialogflow, RASA is an Open Source alternative, which allows to deploy it on a diverse set of platforms and tune it to the user needs. RASA offers a paid "Enterprise plan" with analytics and support from the company, but the user can get full functionality with the free version.

RASA conversational functionality is based in the context of the conversation. Unlike first pattern-recognition chatbots, where the engine searched for specific word sequences to give a response but did not take in consideration the surrounding information of the conversation, contextual assistants can handle a conversation in a more human-like way, making the correct questions to get the information needed to fulfill more complex needs.

RASA's architecture is modular, allowing to easily integrate with other systems. As it is coded in Python and makes use of standard libraries such as *scikit-learn* or *spaCy*, its parts can be reutilized in other projects and implementations. Also, RASA allows the user to override the built-in libraries and scripts used and use their own, giving assistants more flexibility in the development than their competitors.

The main components of RASA are the following:

- **RASA NLU:** *Natural Language Processing* module used in the interpreter. It is the part where the messages sent from the user are processed. It can be trained in any language by Operations like *tokenisation*, *Parts Of Speech* (POS) anotation or *featurizing* are made in order to process the entry. These operations are performed through a predefined pipeline that the user can configure to its needs.

RASA NLU extract the intent that matches the message received from the user, the entities that might be used in the responses generation and any other structured information.

- **RASA Core:** Core is the dialogue engine used for building AI assistants. It is built based in machine learning models trained with conversations to decide the actions to perform, instead of classic programming if/else statements.

It is based on *stories*, which are representations of a conversation between the user and the assistant. There, the intents are related to specific action names, so RASA Core can process the input data and give the correct answers. As other assistants, RASA can also connect with external services and events.

To train the model, the administrator can train it by conversation providing feedback while having a conversation. This can be made through command line or using RASA X, a toolkit with user interface to collect conversations of the assistant and train it from the stored information[7].

In the figure 3.6, the whole message processing is shown:

The steps followed in processing the message are:

- The message is received and passed to the Interpreter. It performs several transformations and operations to convert it into a dictionary with the original text, the intent and entities found in it.
- The object that keeps track of the conversation state is the Tracker, which receives the info that a new message has arrived

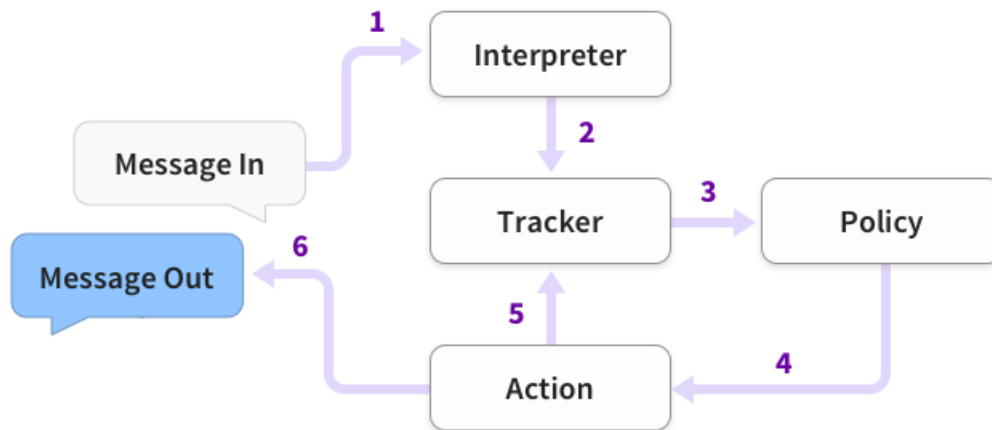


Figure 3.6: Incoming message processing [5]

- The policy receives the current state of the tracker, and in consequence chooses decides which action to take next.
- The chosen action is logged by the tracker, and then sent to the user

### 3.4 Conclusions

In this chapter we have introduced some of the main technologies used in this project and some of the options that we may use.

First of all, we have described the cognitive systems, which is one of the basis of chatbots. We have shown some of the main uses for these systems, as well as some of the most important properties that conform cognitive systems.

After that, we talked about chatbots, gathering information about their origins and the first applications of the technology. After that, we described a brief classification about the different parts that conforms a chatbot, and the task that they perform. Also, we described some of the main frameworks to build chatterbots, presented their main properties and how they work.



## Architecture and Methodology

---

*This chapter presents the methodology used in this work. It describes the overall architecture of the project, with the connections between the different components involved on the development of the project.*

## **4.1 Methodology**

## 4.2 Architecture





# CHAPTER 5

## Case study

---

*In this chapter we are going to describe a selected use case. This section will discuss the two use case: Sentiment and Emotion Analysis applying Deep Learning techniques, which will include the datasets used, the first steps with the neural networks and the use of more advance techniques.*

## **5.1 Techniques**

## **5.2 Datasets**

## **5.3 Results**

...

# CHAPTER 6

## Conclusions

---

*This chapter will describe the achieved goals done by the master thesis following some the key points developed in the project.*

## 6.1 Achieved Goals

The achieved goals for this project are the following ones:

- ...

## 6.2 Conclusion

The project has fulfilled the proposed objectives, but we must refer to each one of these objectives in to draw a general conclusion.



## APPENDIX **A**

### Example

---

This is the description of the appendix.





# Bibliography

---

- [1] Chatbots in customer service - statistics and trends [infographic]. <https://www.invespcro.com/blog/chatbots-customer-service/>.
- [2] Cognitive computing defined – cognitive computing consortium. <https://cognitivecomputingconsortium.com/resources/cognitive-computing-defined/#1467829079735-c0934399-599a>.
- [3] The deepqa research team - ibm. [https://researcher.watson.ibm.com/researcher/view\\_group.php?id=2099](https://researcher.watson.ibm.com/researcher/view_group.php?id=2099).
- [4] Dialogflow use cases. <https://dialogflow.com/case-studies/dominos/>.
- [5] Dialogflow — google cloud. <https://cloud.google.com/dialogflow>.
- [6] Fulfillment — dialogflow documentation — google cloud. <https://cloud.google.com/dialogflow/docs/fulfillment-overview>, month = , year = ,.
- [7] Improve your contextual assistant with rasa x. <https://rasa.com/docs/rasa-x/>.
- [8] Microsoft hololens — mixed reality technology for business, <https://www.microsoft.com/en-gb/hololens>.
- [9] Microsoft hololens, <https://www.microsoft.com/en-gb/hololens>.
- [10] Sameera A Abdul-Kader and JC Woods. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7), 2015.
- [11] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [12] Ignacio Corcuera-Platas. Development of a Deep Learning Based Sentiment Analysis and Evaluation Service. Master thesis, ETSI Telecomunicación, Madrid, January 2018.
- [13] J. O. Gutierrez-Garcia and E. López-Neri. Cognitive computing: A brief survey and open research challenges. In *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pages 328–333, 2015.
- [14] IBM. Watson assistant for customer self-service. [https://www.ibm.com/watson/assets/duo/pdf/190905\\_WatsonAssistant\\_ChildPagePDF\\_CSS\\_.pdf](https://www.ibm.com/watson/assets/duo/pdf/190905_WatsonAssistant_ChildPagePDF_CSS_.pdf).

- [15] Lorenz Cuno Klopfenstein, Saverio Delpriori, Silvia Malatini, and Alessandro Bogliolo. The rise of bots: A survey of conversational interfaces, patterns, and paradigms. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS '17, page 555–565, New York, NY, USA, 2017. Association for Computing Machinery.
- [16] Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. *Minds and machines*, 10(4):463–518, 2000.
- [17] Heung-Yeung Shum, Xiao-dong He, and Di Li. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26, 2018.